This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

OIPE JC596
JUN 0 2 2004
PATENT & TRADEMARK OFFICE

Figures

Figure 1

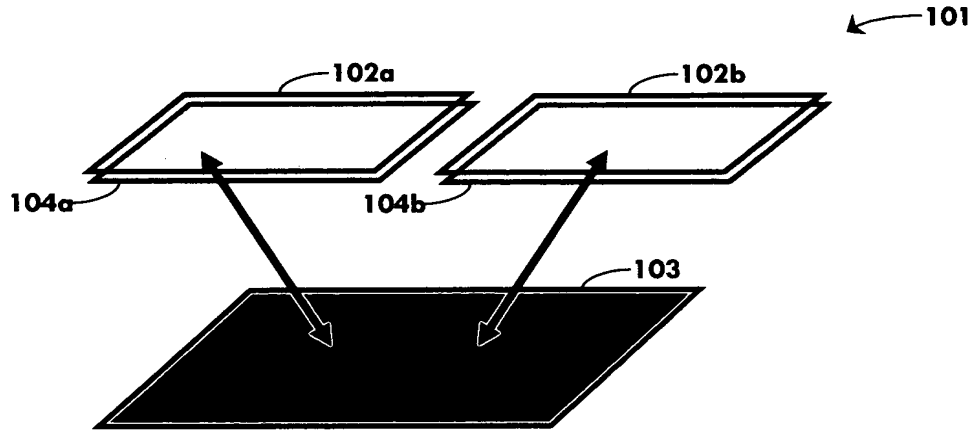


Figure 2

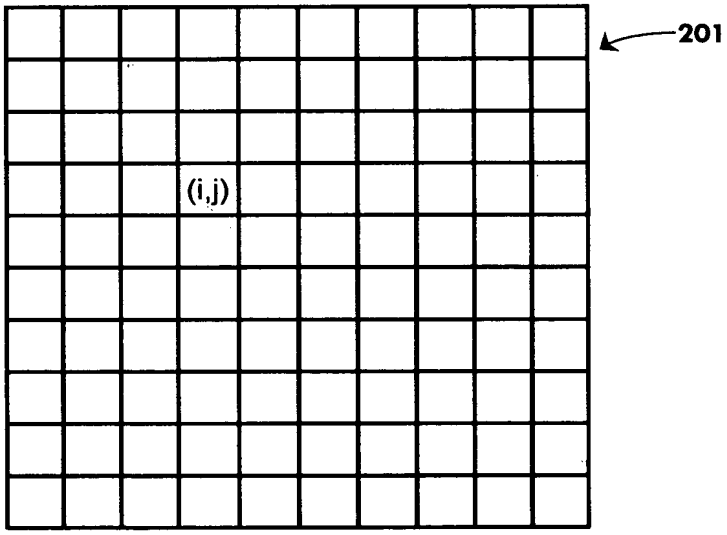


Figure 3

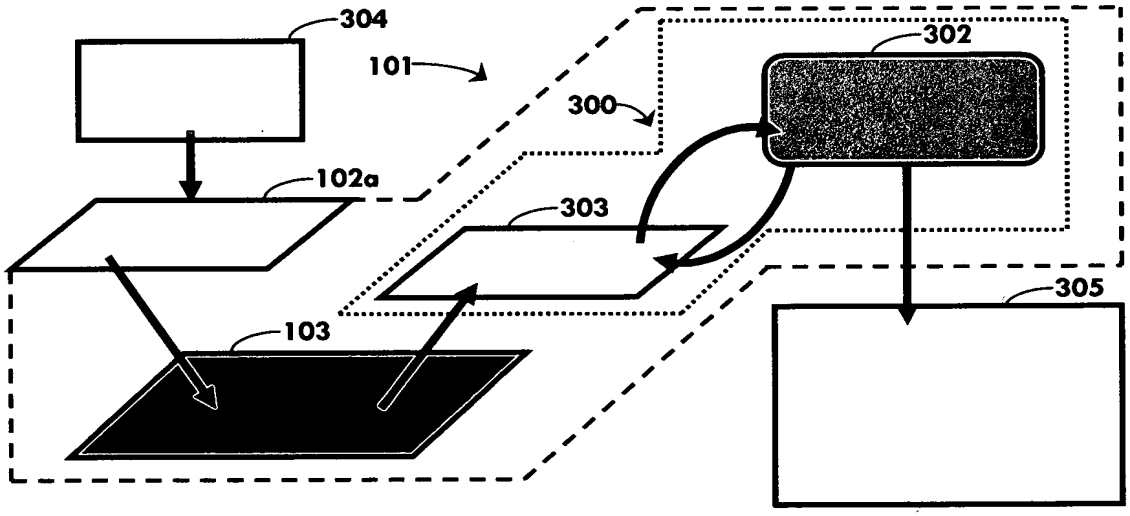


Figure 4

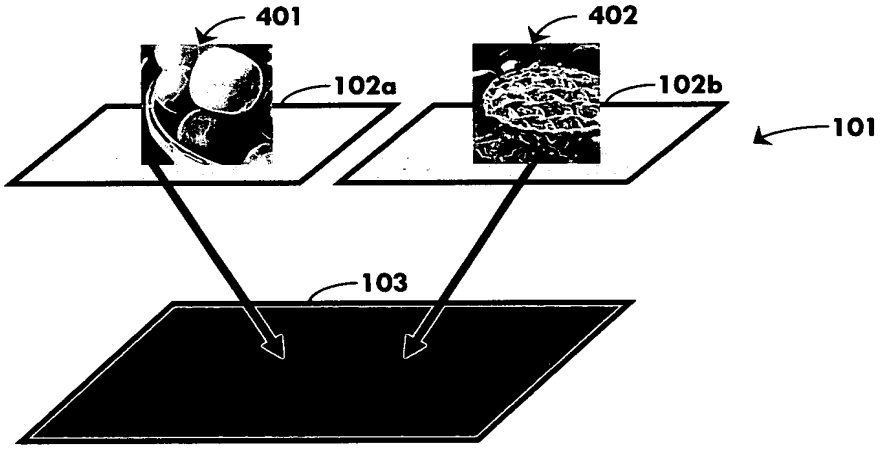


Figure 5

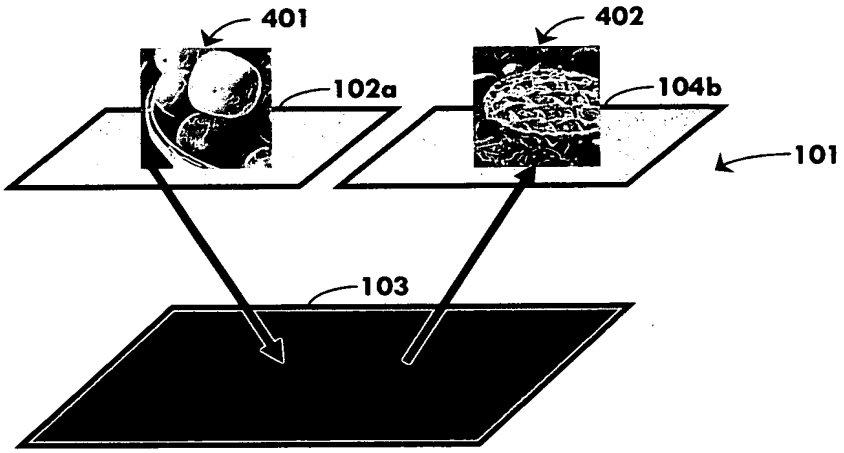


Figure 6

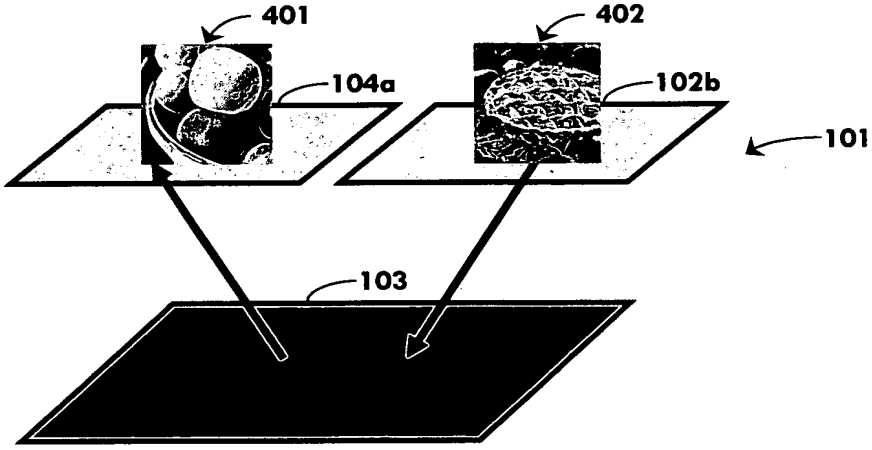


Figure 7

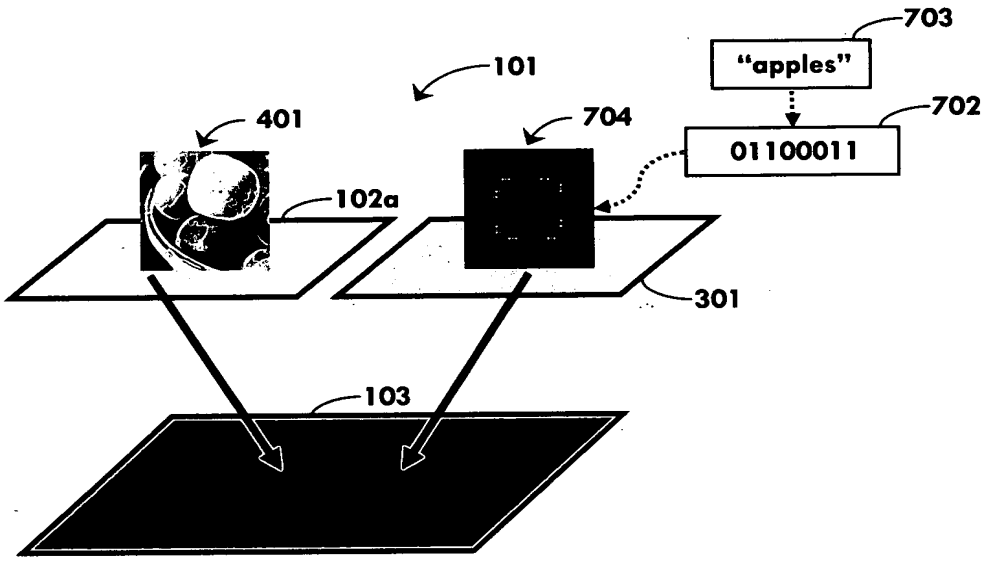


Figure 8

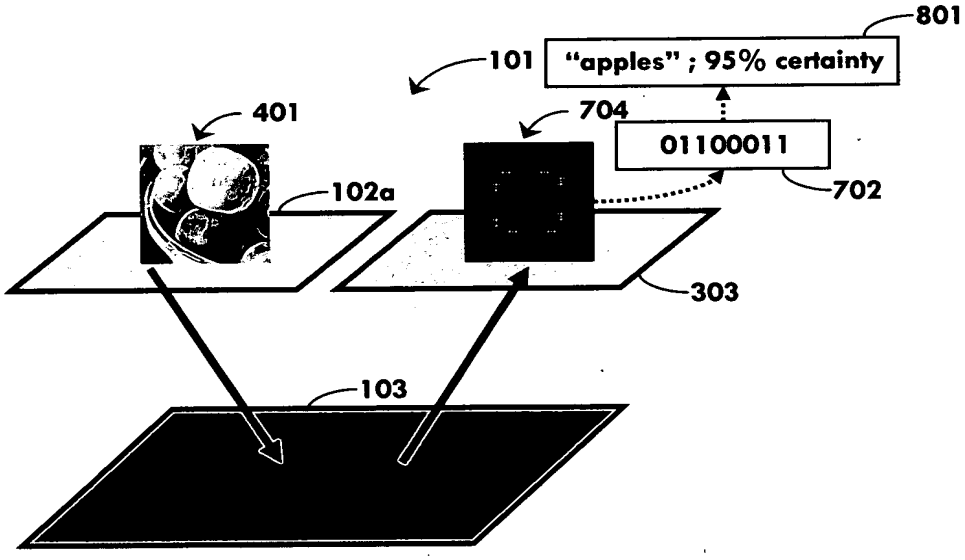


Figure 9

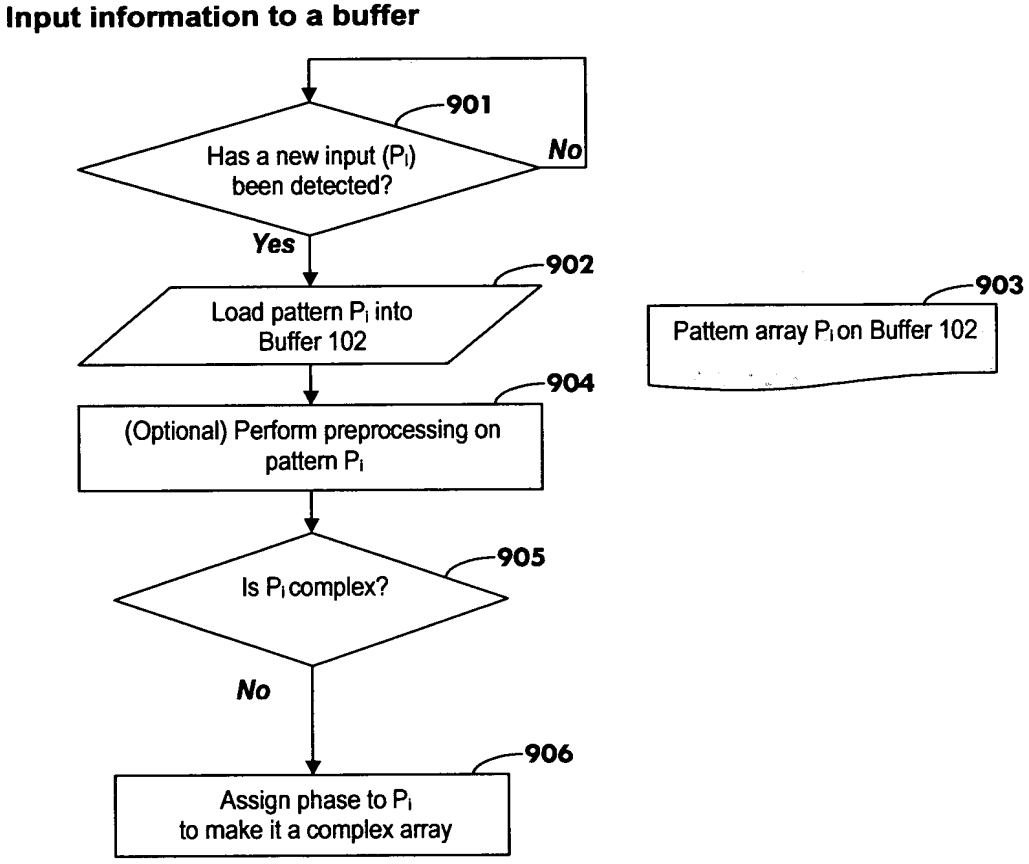

Input information to a buffer
901
Has a new input (P_i) been detected?
No
Yes
902
Load pattern P_i into Buffer 102
903
Pattern array P_i on Buffer 102
904
(Optional) Perform preprocessing on pattern P_i
905
Is P_i complex?
No
906
Assign phase to P_i to make it a complex array

Figure 10

Transport information from buffer to cortex or cortex to buffer

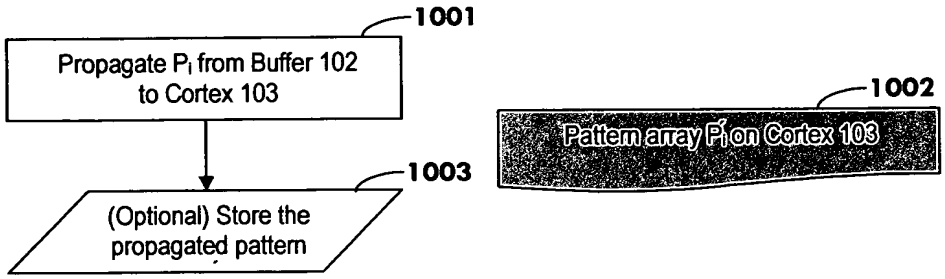


Figure 11

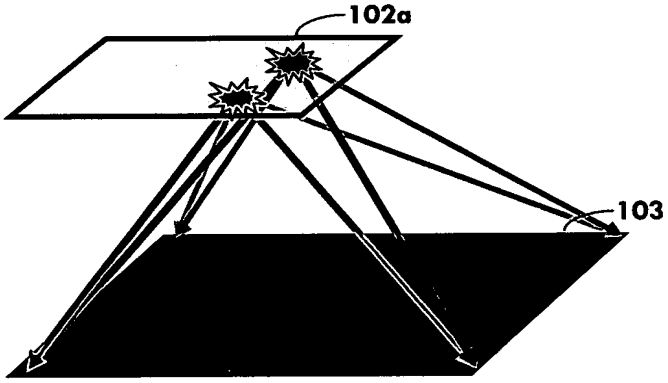


Figure 12

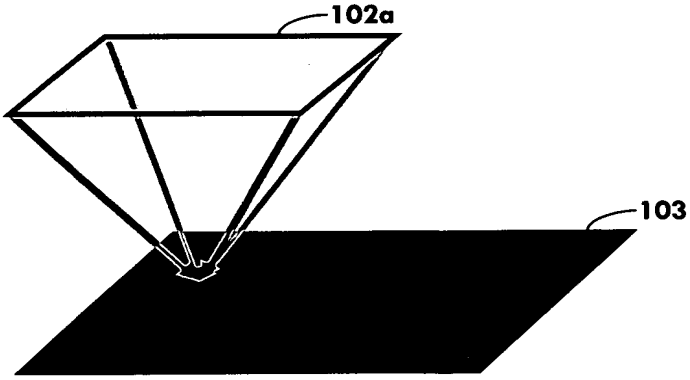


Figure 13

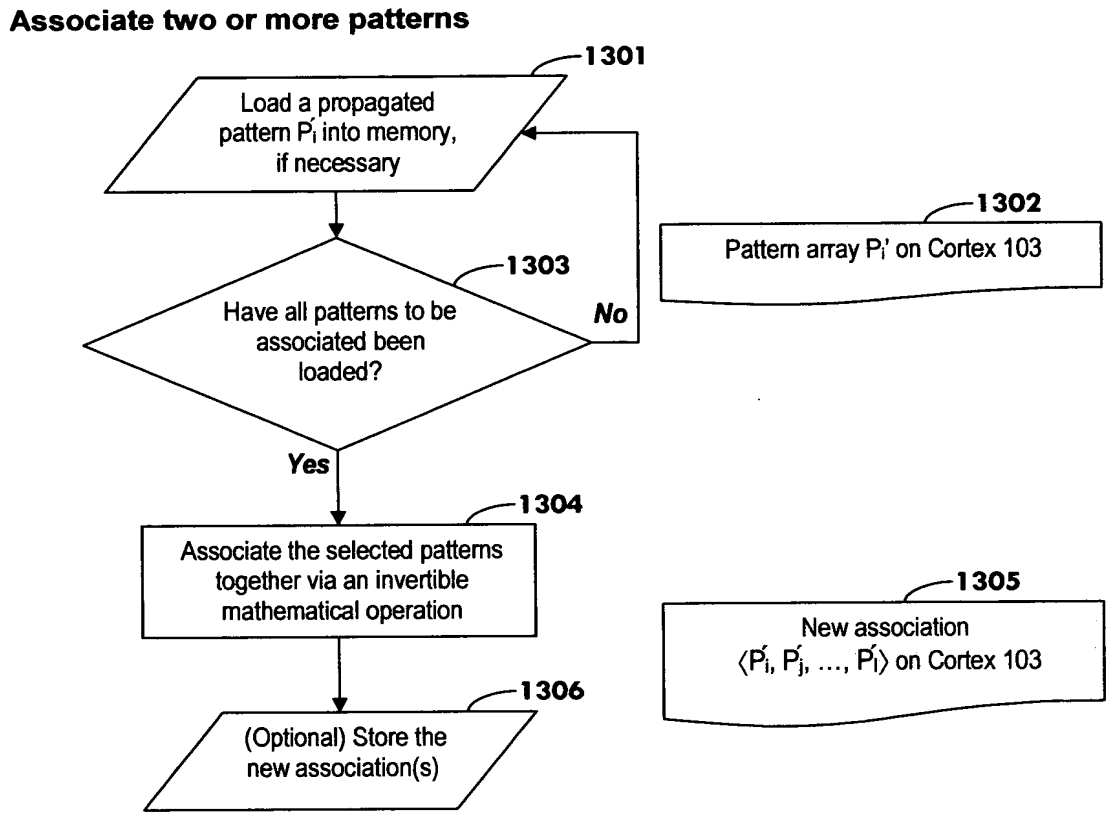

Associate two or more patterns
1301
Load a propagated pattern P'i into memory, if necessary
1302
Pattern array Pi' on Cortex 103
1303
Have all patterns to be associated been loaded?
No
Yes
1304
Associate the selected patterns together via an invertible mathematical operation
1305
New association ⟨P'i, P'j, ..., P'l⟩ on Cortex 103
1306
(Optional) Store the new association(s)

Figure 14

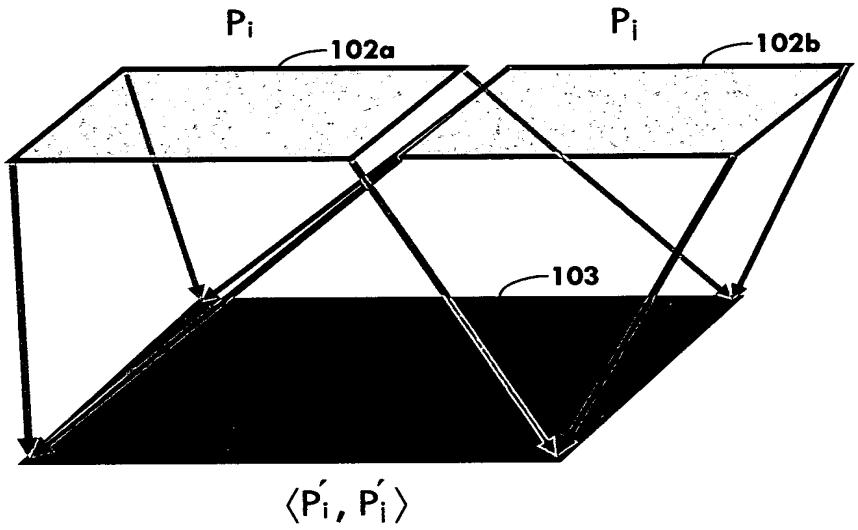


Figure 15

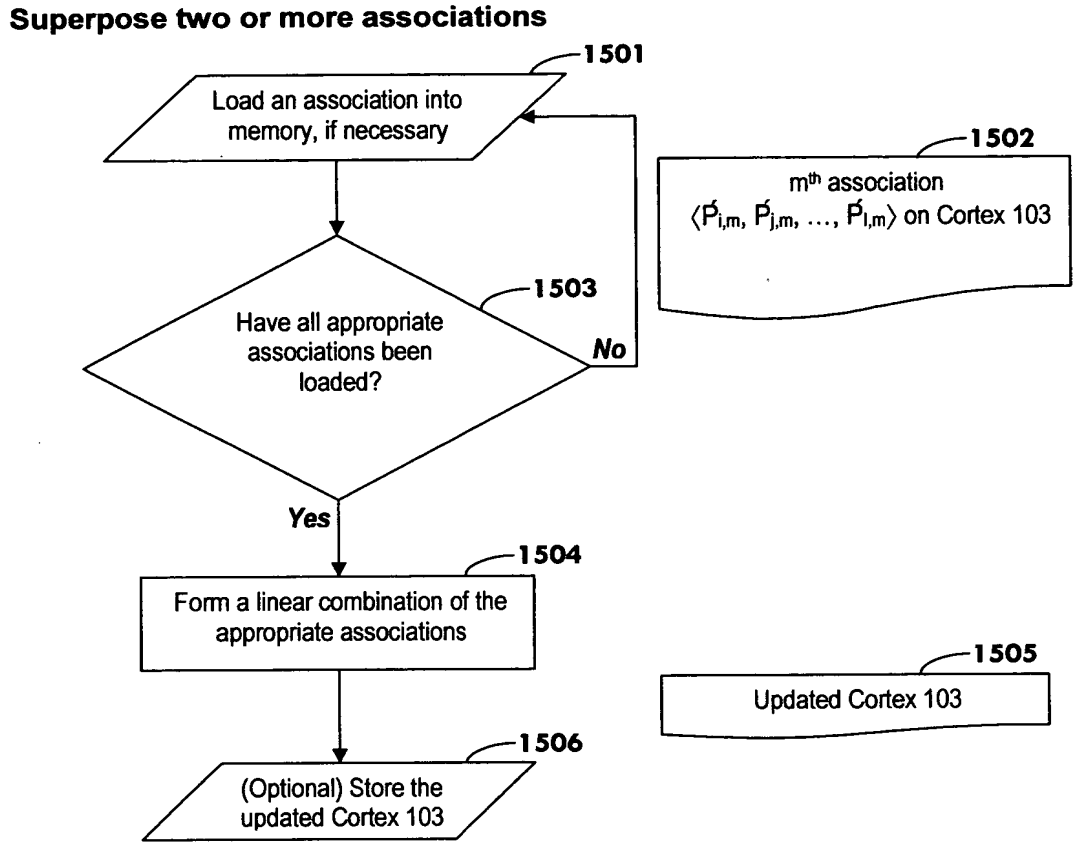

Superpose two or more associations
1501
Load an association into memory, if necessary
1502
m^th association
⟨Ṕi,m, Ṕj,m, ..., Ṕl,m⟩ on Cortex 103
1503
Have all appropriate associations been loaded?
No
Yes
1504
Form a linear combination of the appropriate associations
1505
Updated Cortex 103
1506
(Optional) Store the updated Cortex 103

Figure 16

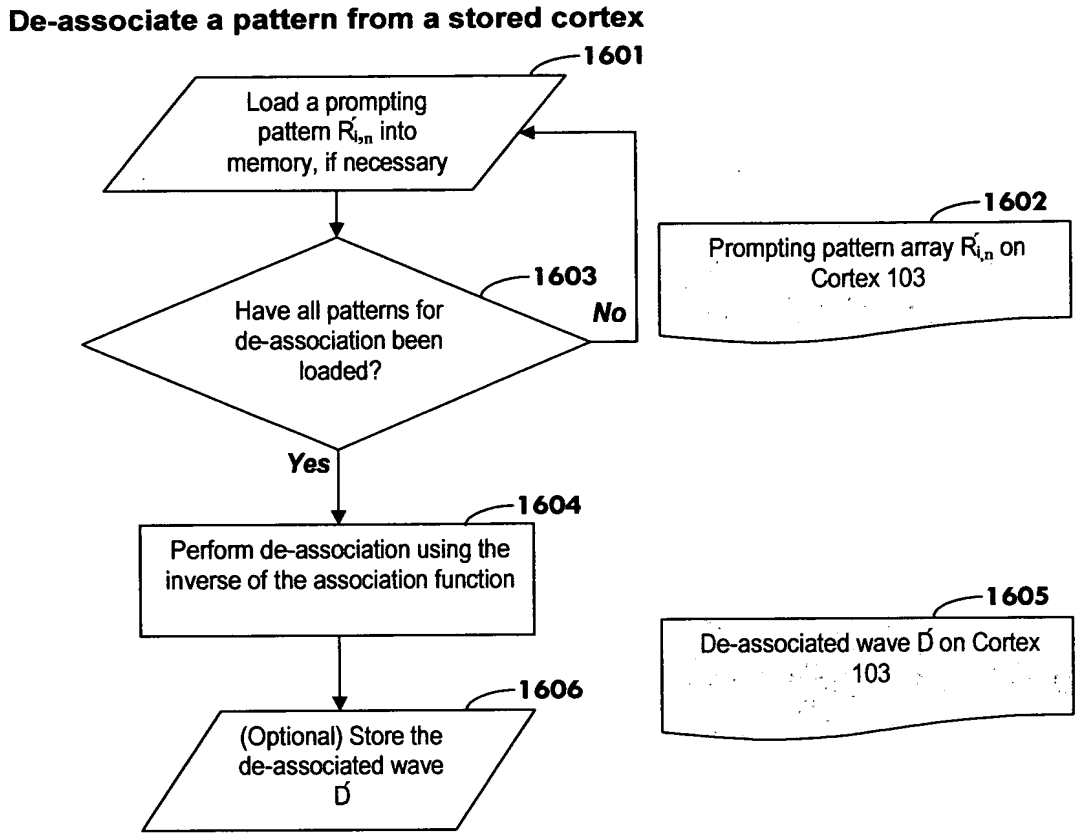

De-associate a pattern from a stored cortex
1601
Load a prompting pattern $\acute{R}_{i,n}$ into memory, if necessary
1602
Prompting pattern array $\acute{R}_{i,n}$ on Cortex 103
1603
Have all patterns for de-association been loaded?
No
Yes
1604
Perform de-association using the inverse of the association function
1605
De-associated wave $\acute{D}$ on Cortex 103
1606
(Optional) Store the de-associated wave $\acute{D}$

Figure 17

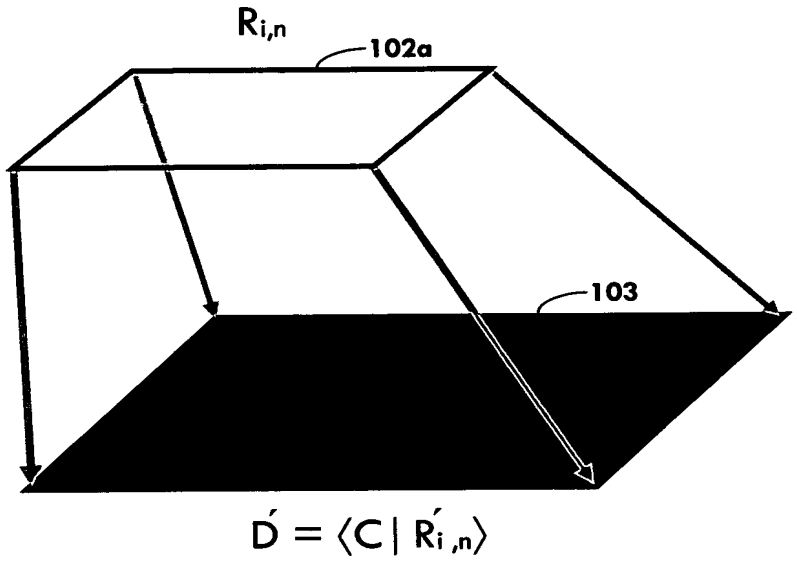


Figure 18

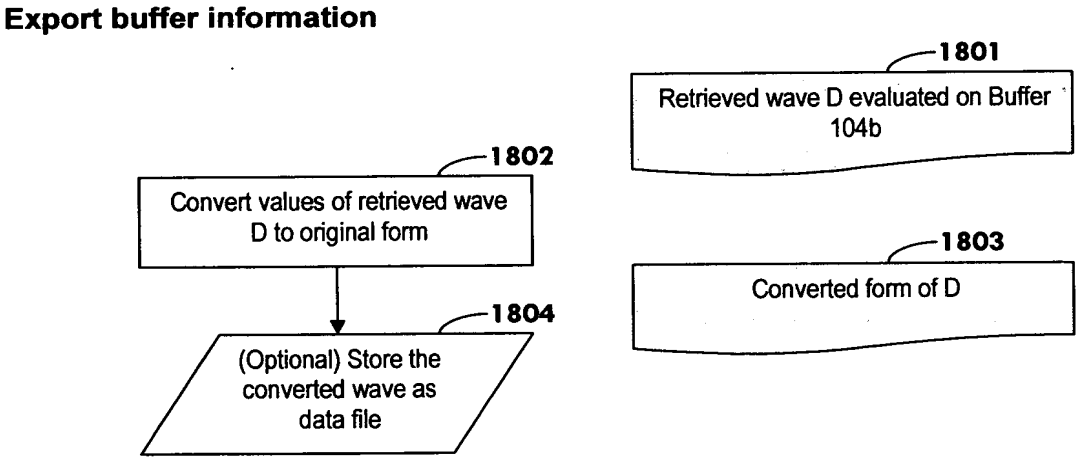


Figure 19

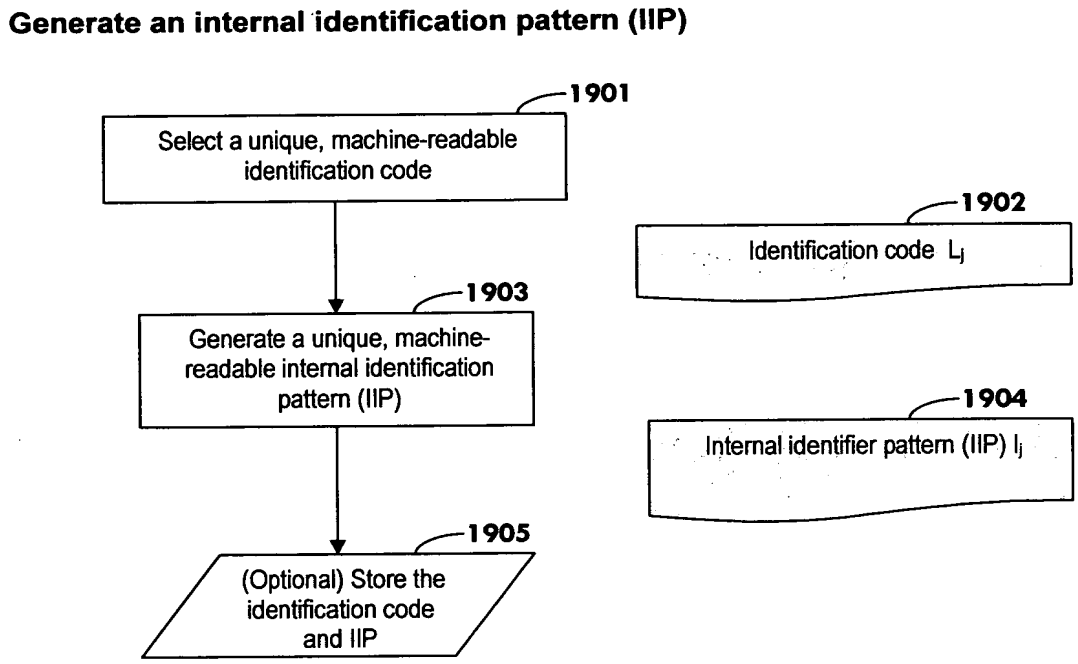


Figure 20

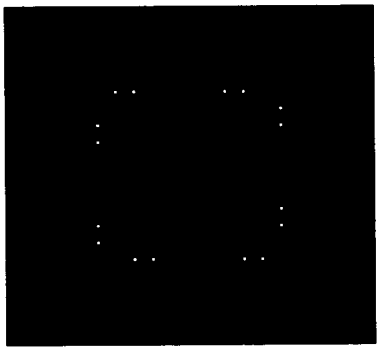


Figure 21

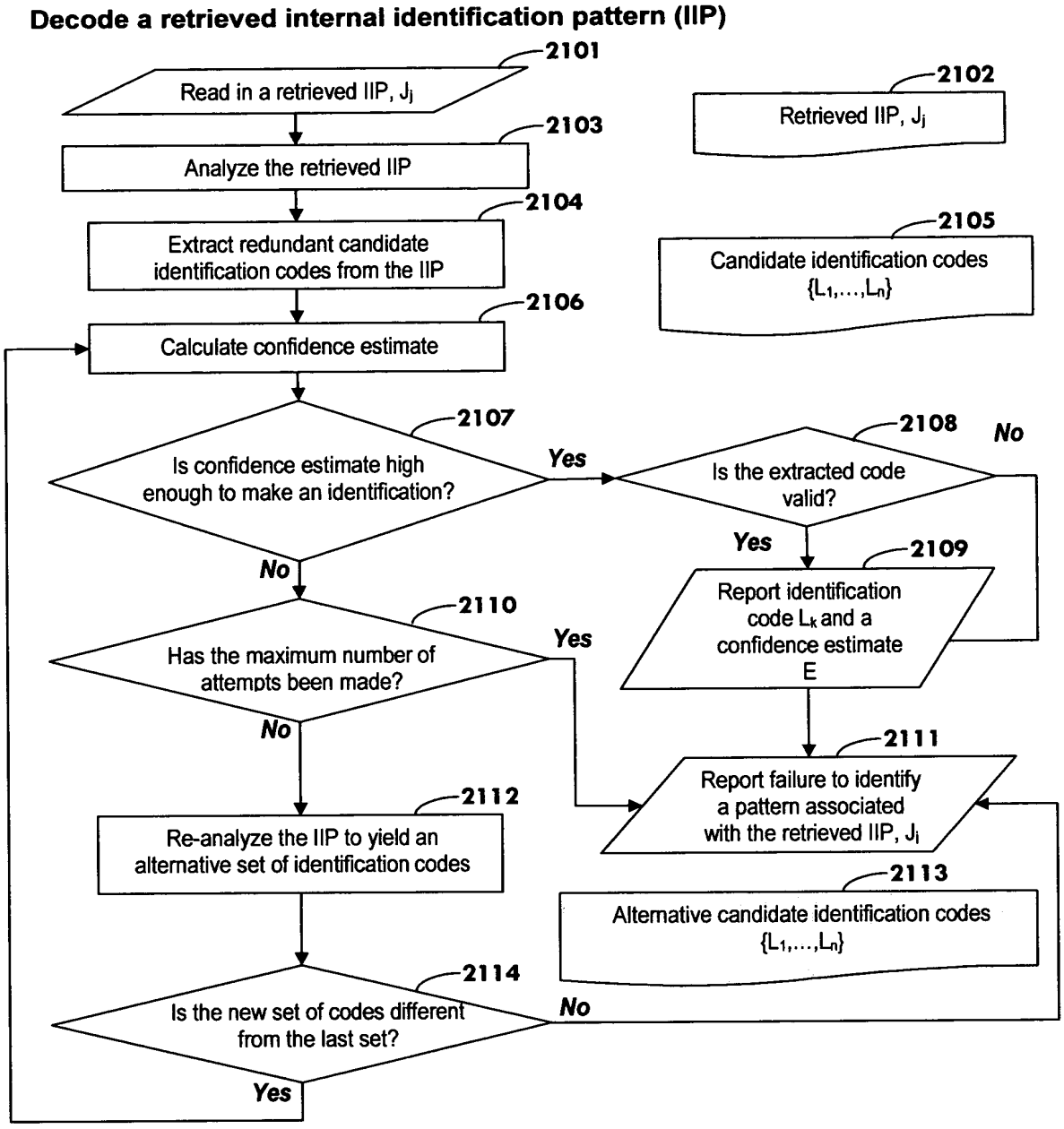

Decode a retrieved internal identification pattern (IIP)
2101
Read in a retrieved IIP, Jj
2102
Retrieved IIP, Jj
2103
Analyze the retrieved IIP
2104
Extract redundant candidate identification codes from the IIP
2105
Candidate identification codes {L1,...,Ln}
2106
Calculate confidence estimate
2107
Is confidence estimate high enough to make an identification?
Yes
2108
Is the extracted code valid?
No
Yes
2109
Report identification code Lk and a confidence estimate E
No
2110
Has the maximum number of attempts been made?
Yes
No
2111
Report failure to identify a pattern associated with the retrieved IIP, Ji
2112
Re-analyze the IIP to yield an alternative set of identification codes
2113
Alternative candidate identification codes {L1,...,Ln}
2114
Is the new set of codes different from the last set?
No
Yes

Figure 22

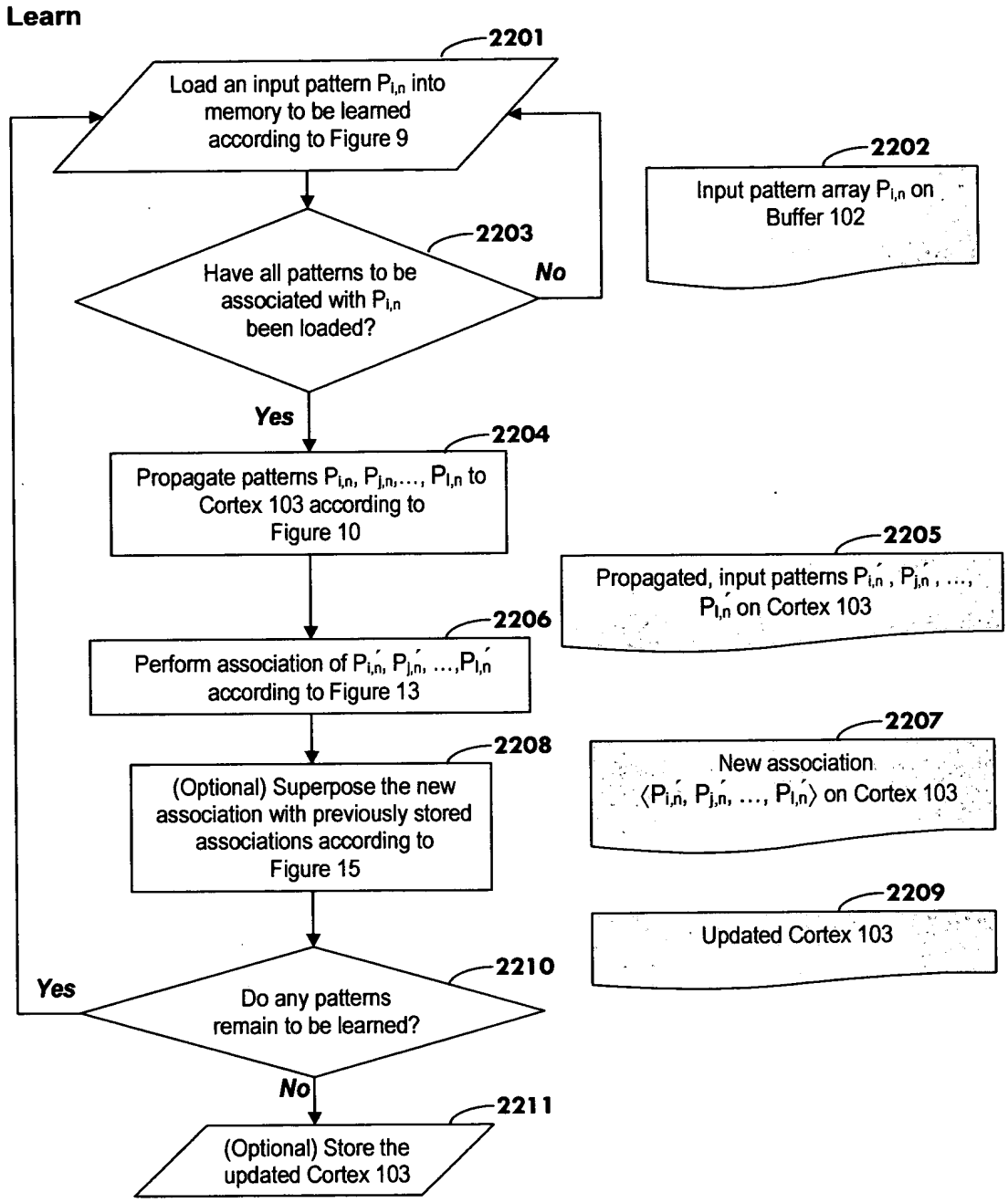

Learn
2201
Load an input pattern $P_{i,n}$ into memory to be learned according to Figure 9
2202
Input pattern array $P_{i,n}$ on Buffer 102
2203
Have all patterns to be associated with $P_{i,n}$ been loaded?
No
Yes
2204
Propagate patterns $P_{i,n}$, $P_{j,n}$,..., $P_{l,n}$ to Cortex 103 according to Figure 10
2205
Propagated, input patterns $P_{i,n}'$, $P_{j,n}'$, ..., $P_{l,n}'$ on Cortex 103
2206
Perform association of $P_{i,n}'$, $P_{j,n}'$, ...,$P_{l,n}'$ according to Figure 13
2207
New association $\langle P_{i,n}', P_{j,n}', \ldots, P_{l,n}' \rangle$ on Cortex 103
2208
(Optional) Superpose the new association with previously stored associations according to Figure 15
2209
Updated Cortex 103
2210
Yes
Do any patterns remain to be learned?
No
2211
(Optional) Store the updated Cortex 103

Figure 23

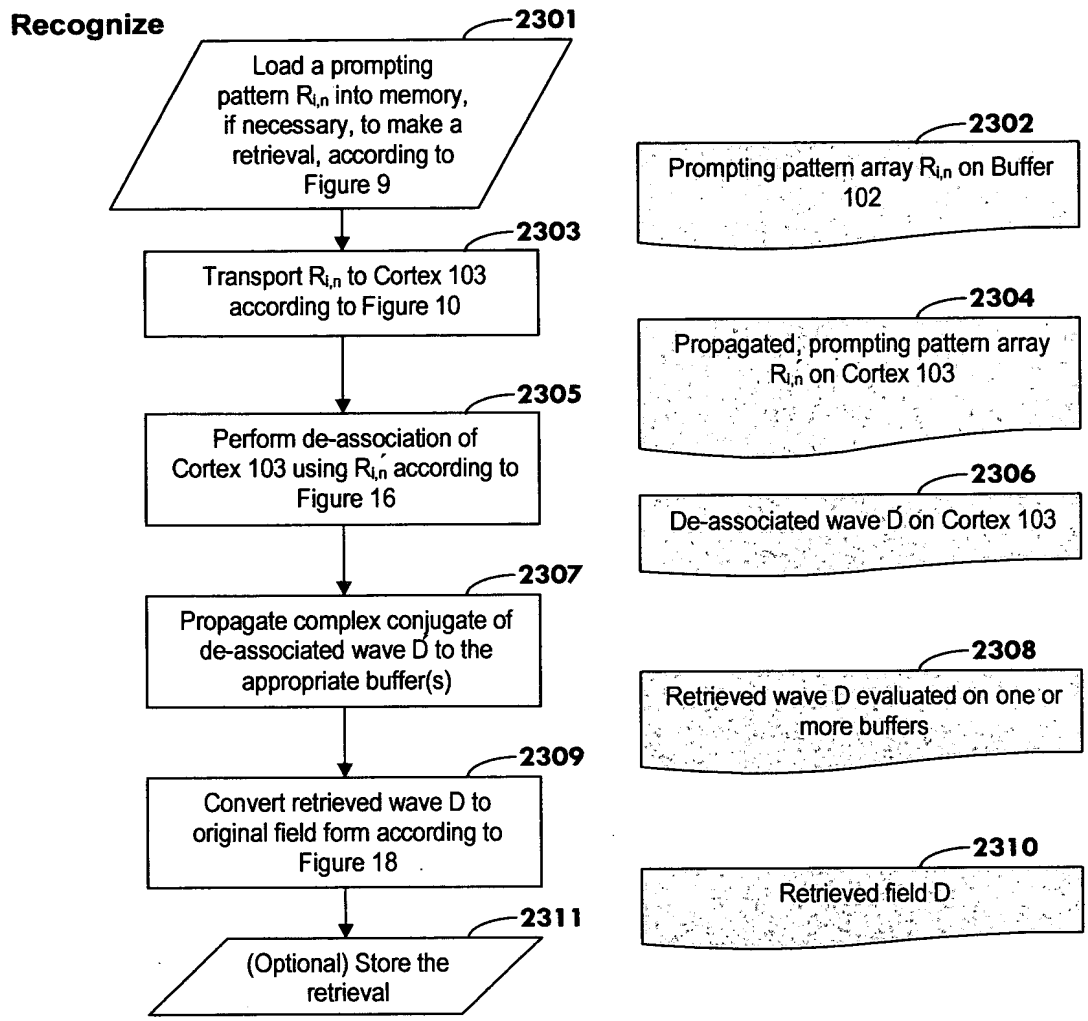

Recognize
2301
Load a prompting pattern $R_{i,n}$ into memory, if necessary, to make a retrieval, according to Figure 9
2303
Transport $R_{i,n}$ to Cortex 103 according to Figure 10
2305
Perform de-association of Cortex 103 using $R_{i,n}$ according to Figure 16
2307
Propagate complex conjugate of de-associated wave $\acute{D}$ to the appropriate buffer(s)
2309
Convert retrieved wave D to original field form according to Figure 18
2311
(Optional) Store the retrieval
2302
Prompting pattern array $R_{i,n}$ on Buffer 102
2304
Propagated, prompting pattern array $R_{i,n}$ on Cortex 103
2306
De-associated wave $\acute{D}$ on Cortex 103
2308
Retrieved wave D evaluated on one or more buffers
2310
Retrieved field D

Figure 24

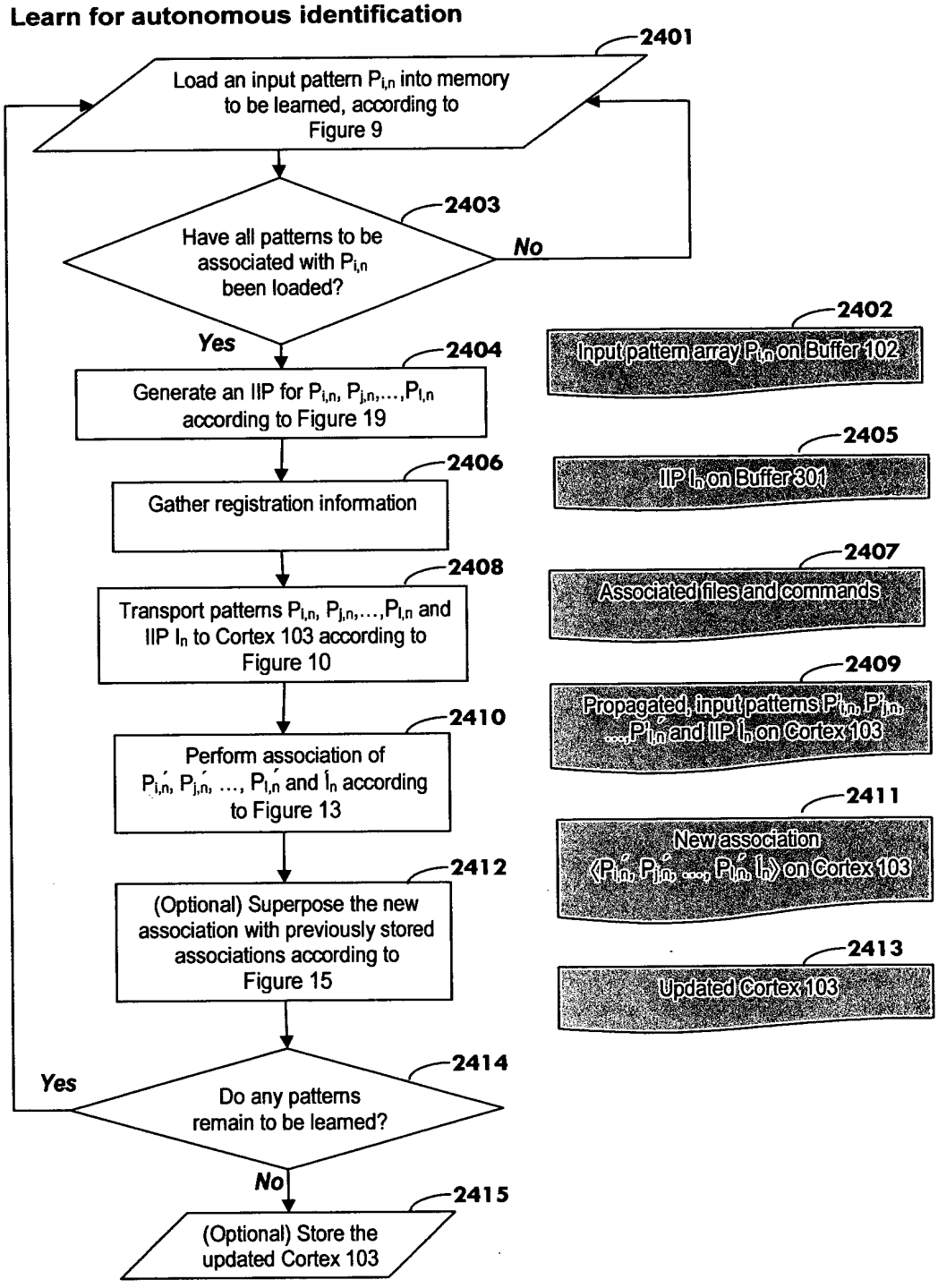

Learn for autonomous identification
2401
Load an input pattern $P_{i,n}$ into memory to be learned, according to Figure 9
2403
Have all patterns to be associated with $P_{i,n}$ been loaded?
No
Yes
2404
Generate an IIP for $P_{i,n}$, $P_{j,n}$,..., $P_{l,n}$ according to Figure 19
2406
Gather registration information
2408
Transport patterns $P_{i,n}$, $P_{j,n}$,..., $P_{l,n}$ and IIP I_n to Cortex 103 according to Figure 10
2410
Perform association of $P'_{i,n}$, $P'_{j,n}$, ..., $P'_{l,n}$ and I_n according to Figure 13
2412
(Optional) Superpose the new association with previously stored associations according to Figure 15
2414
Yes
Do any patterns remain to be learned?
No
2415
(Optional) Store the updated Cortex 103
2402
Input pattern array $P_{i,n}$ on Buffer 102
2405
IIP I_n on Buffer 301
2407
Associated files and commands
2409
Propagated input patterns $P'_{i,n}$, $P'_{j,n}$, ..., $P'_{l,n}$ and IIP I_n on Cortex 103
2411
New association $\langle P'_{i,n}, P'_{j,n}, \ldots, P'_{l,n}, I_n \rangle$ on Cortex 103
2413
Updated Cortex 103

Figure 25

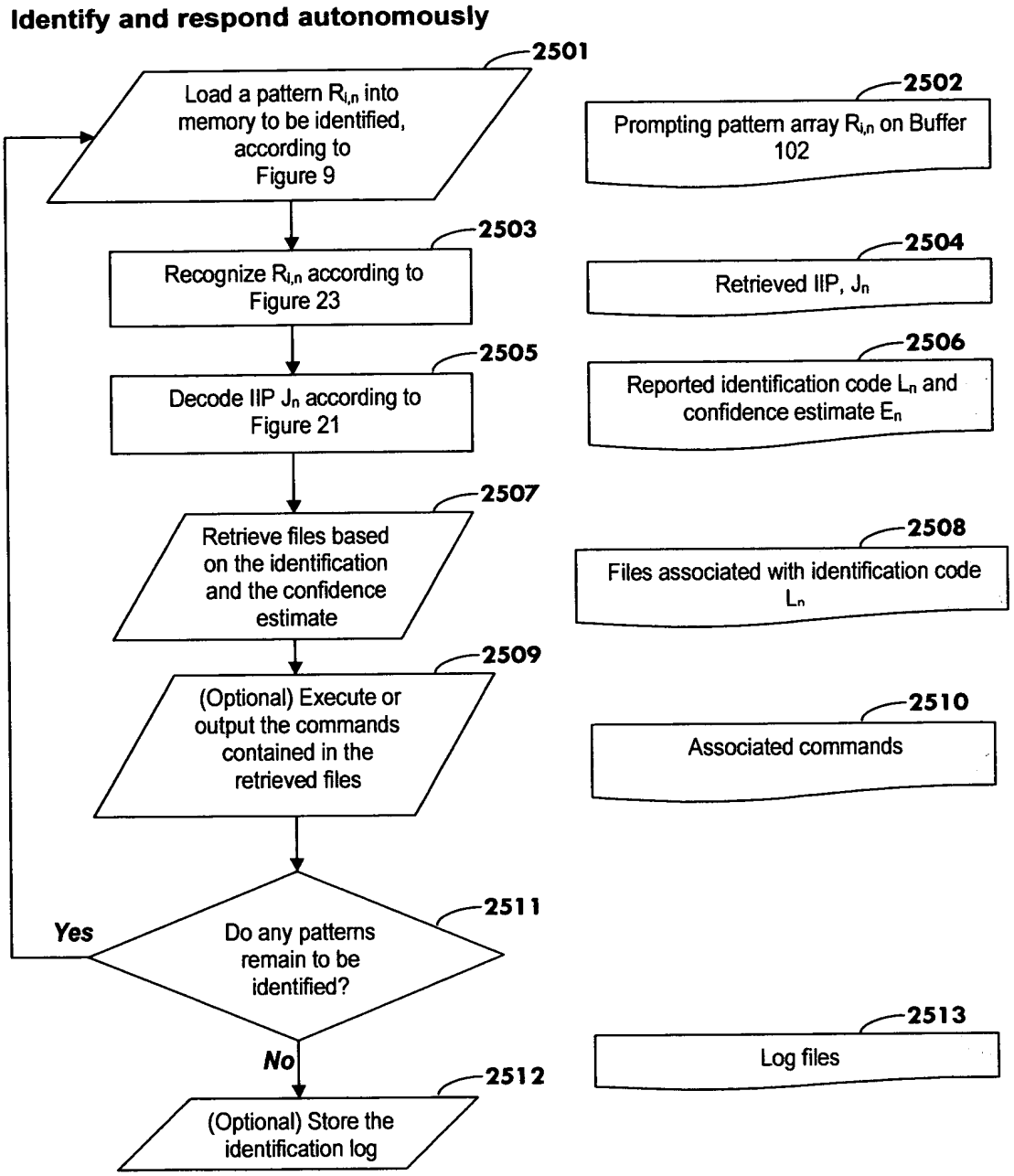

Identify and respond autonomously
2501
Load a pattern $R_{i,n}$ into memory to be identified, according to Figure 9
2502
Prompting pattern array $R_{i,n}$ on Buffer 102
2503
Recognize $R_{i,n}$ according to Figure 23
2504
Retrieved IIP, J_n
2505
Decode IIP J_n according to Figure 21
2506
Reported identification code L_n and confidence estimate E_n
2507
Retrieve files based on the identification and the confidence estimate
2508
Files associated with identification code L_n
2509
(Optional) Execute or output the commands contained in the retrieved files
2510
Associated commands
2511
Do any patterns remain to be identified?
Yes
No
2512
(Optional) Store the identification log
2513
Log files

Figure 26A

Adaptively improve identification performance through experience

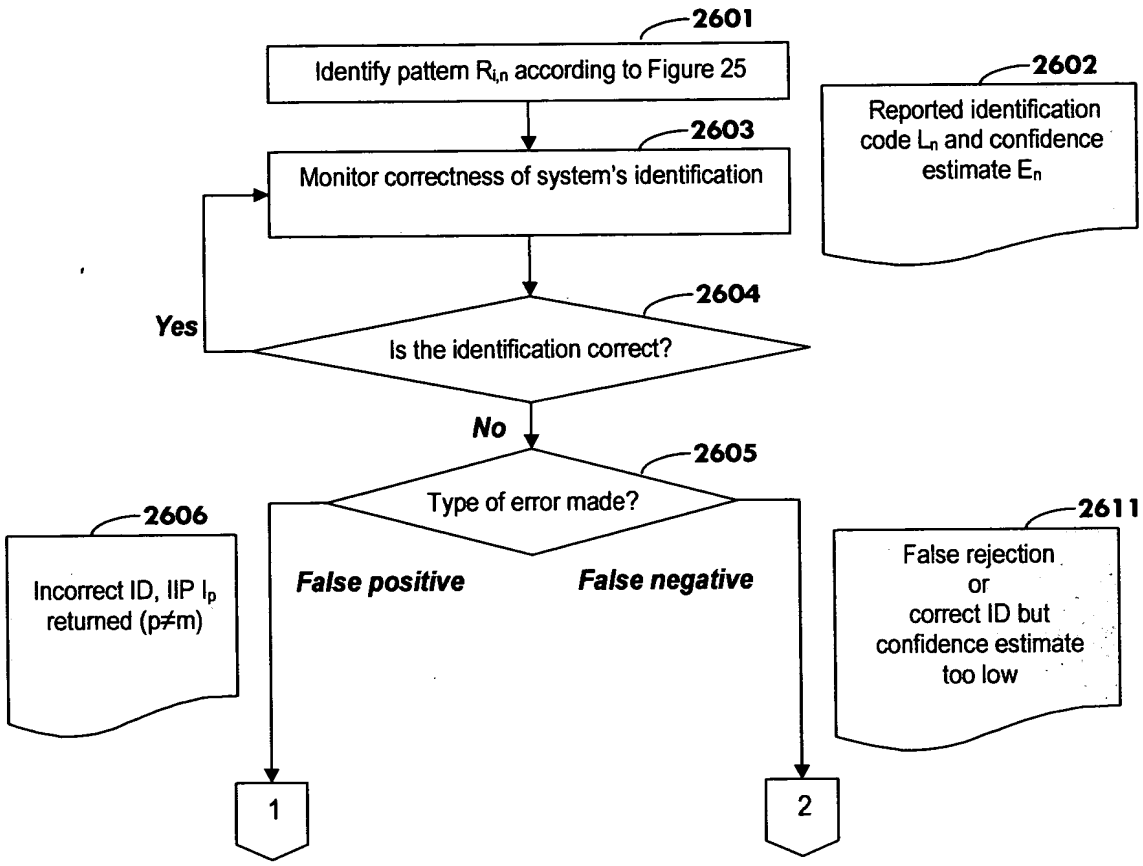


Figure 26B

Adaptively improve identification performance through experience (continued)

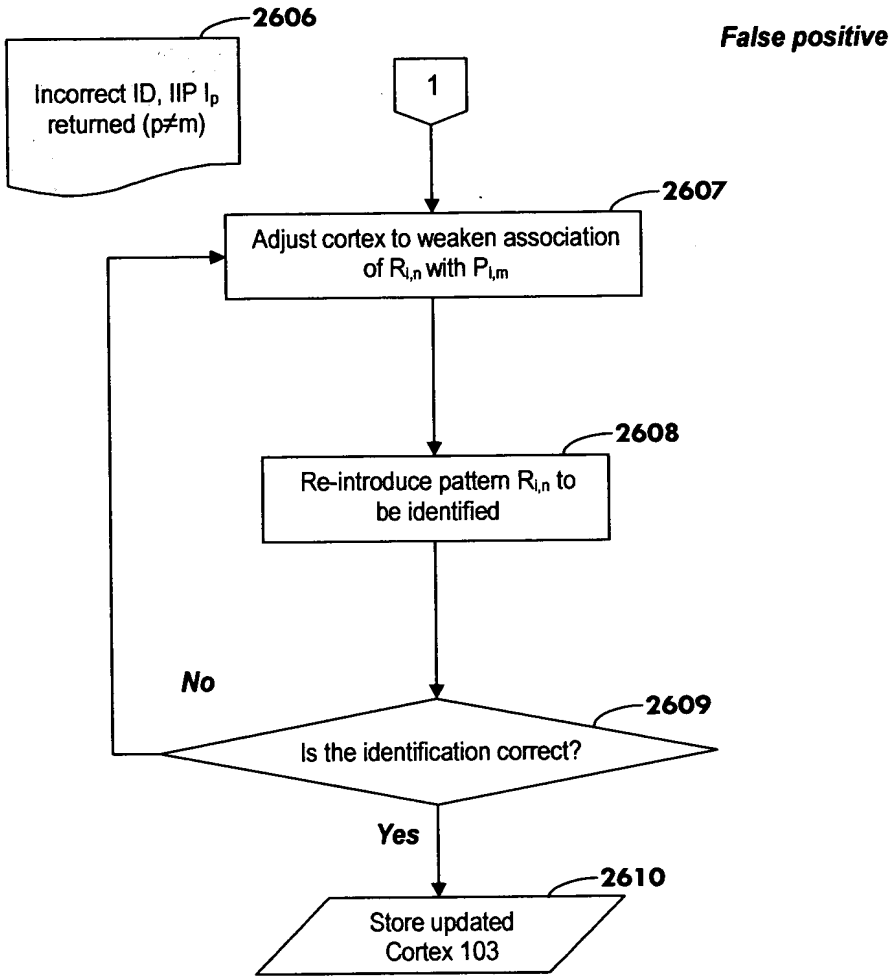
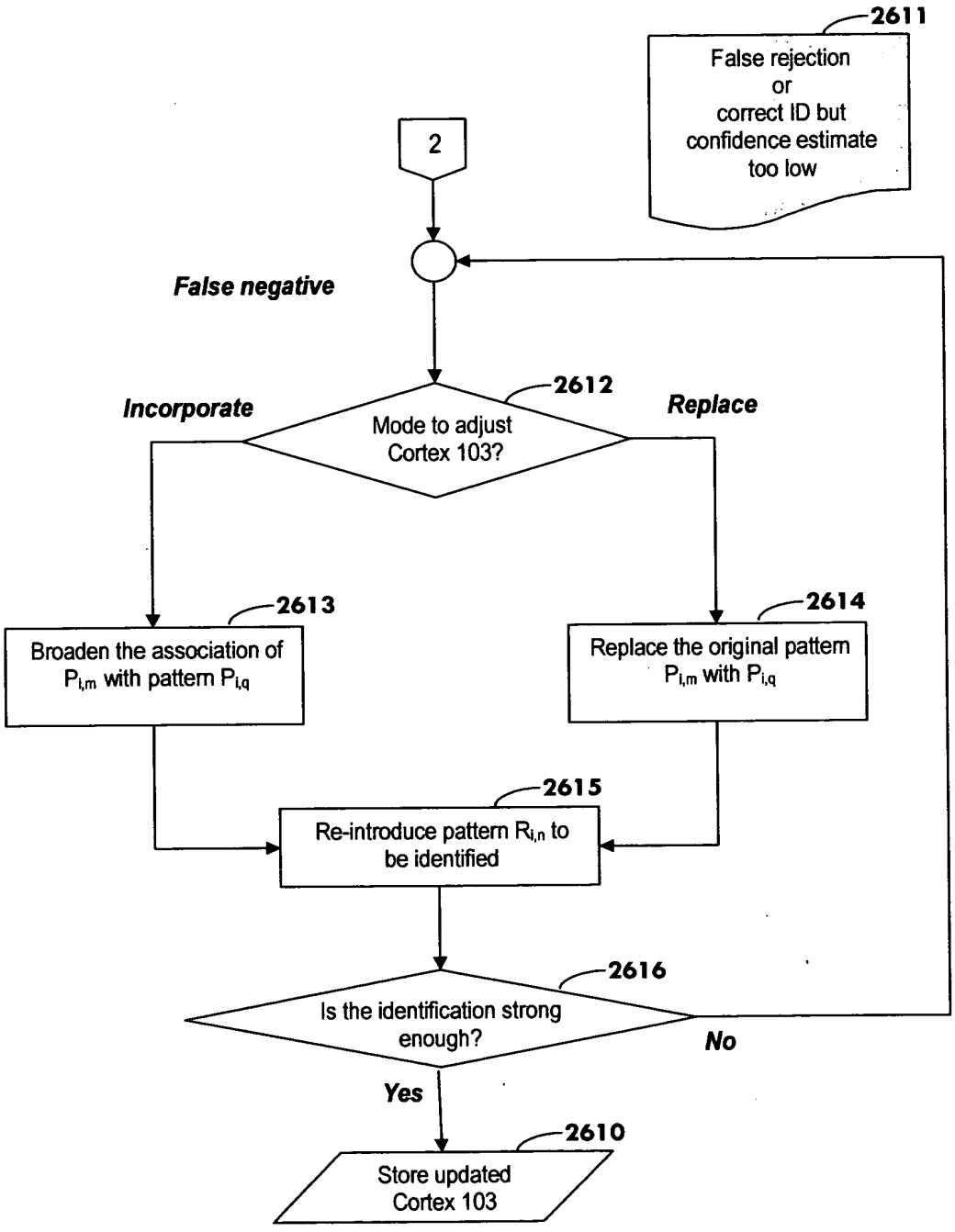


Figure 26C

Adaptively improve identification performance through experience (continued)



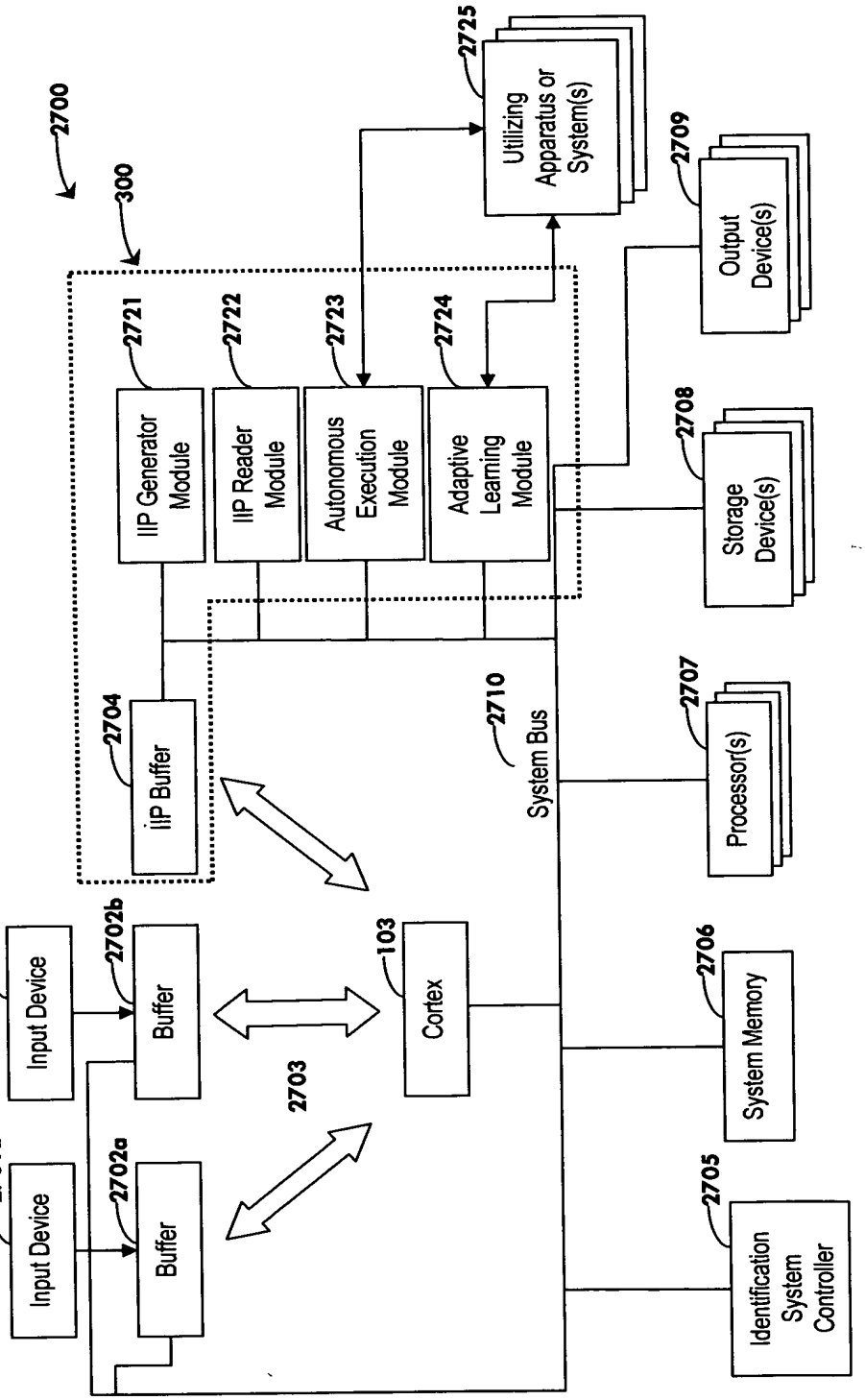

2700
300
2725
Utilizing Apparatus or System(s)
2709
Output Device(s)
2721
IIP Generator Module
2722
IIP Reader Module
2723
Autonomous Execution Module
2724
Adaptive Learning Module
2708
Storage Device(s)
2704
IIP Buffer
2710
System Bus
2707
Processor(s)
2701b
Input Device
2702b
Buffer
103
Cortex
2706
System Memory
2703
2701a
Input Device
2702a
Buffer
2705
Identification System Controller

Figure 27

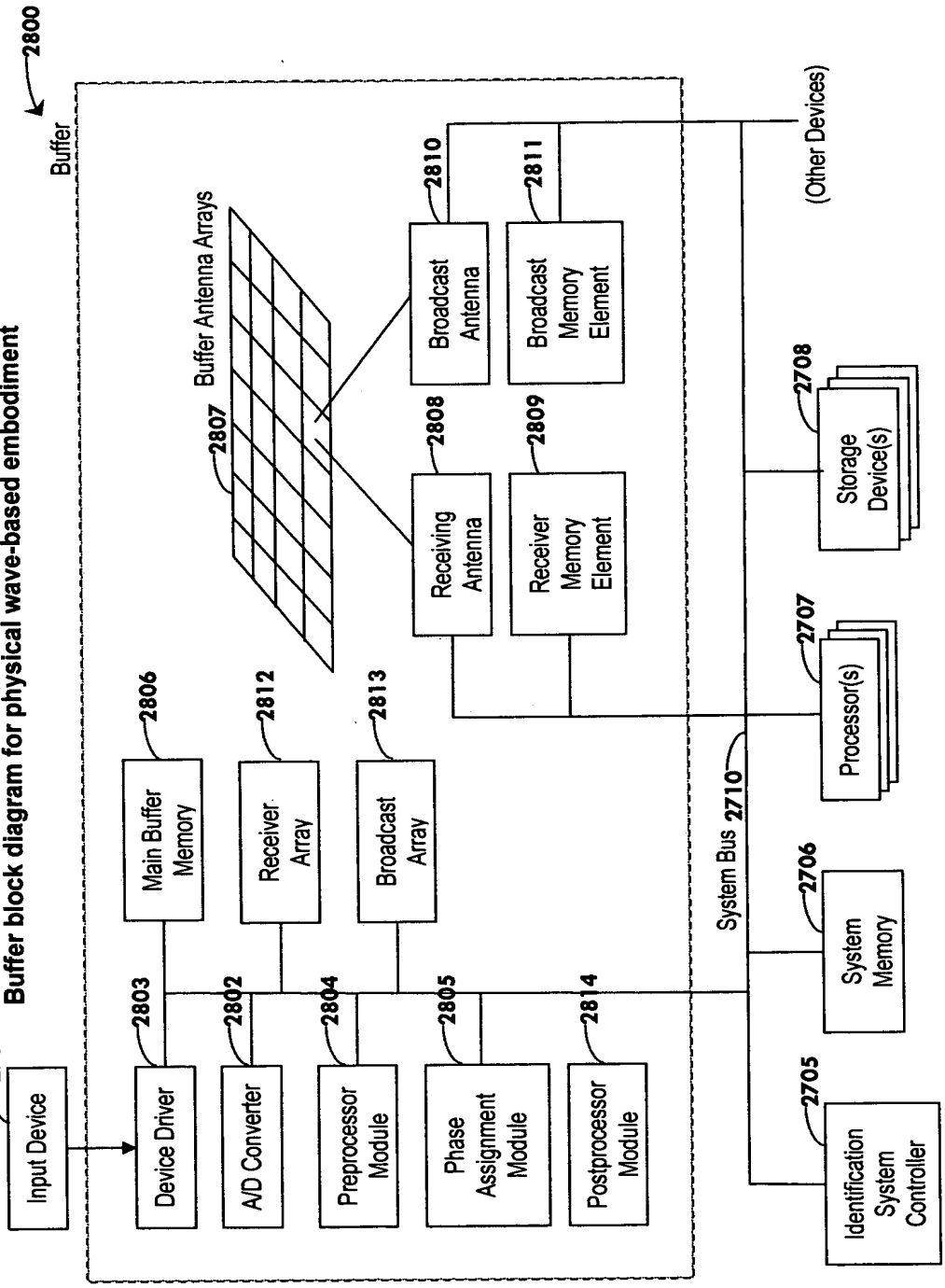

Buffer block diagram for physical wave-based embodiment
2800
Buffer
2701
Input Device
2803
Device Driver
2802
A/D Converter
2804
Preprocessor Module
2805
Phase Assignment Module
2814
Postprocessor Module
2806
Main Buffer Memory
2812
Receiver Array
2813
Broadcast Array
Buffer Antenna Arrays
2807
2808
Receiving Antenna
2809
Receiver Memory Element
2810
Broadcast Antenna
2811
Broadcast Memory Element
System Bus 2710
2705
Identification System Controller
2706
System Memory
2707
Processor(s)
2708
Storage Device(s)
(Other Devices)

Figure 28

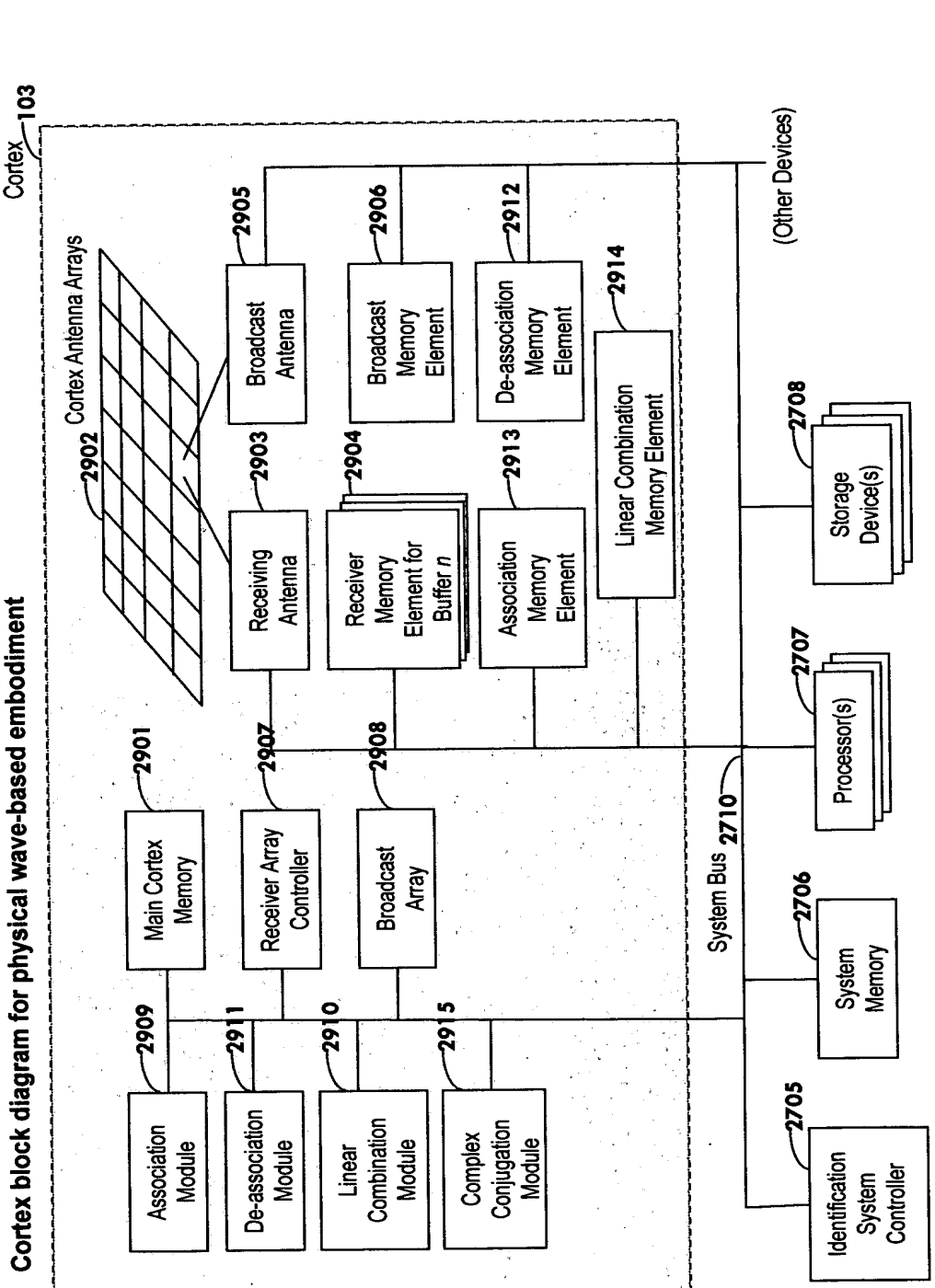

2900
Cortex
103
Cortex block diagram for physical wave-based embodiment
Cortex Antenna Arrays
2902
Broadcast Antenna
2905
Broadcast Memory Element
2906
De-association Memory Element
2912
Linear Combination Memory Element
2914
Receiving Antenna
2903
Receiver Memory Element for Buffer n
2904
Association Memory Element
2913
Storage Device(s)
2708
Processor(s)
2707
Main Cortex Memory
2901
Receiver Array Controller
2907
Broadcast Array
2908
System Bus
2710
(Other Devices)
Association Module
2909
De-association Module
2911
Linear Combination Module
2910
Complex Conjugation Module
2915
System Memory
2706
Identification System Controller
2705

Figure 29

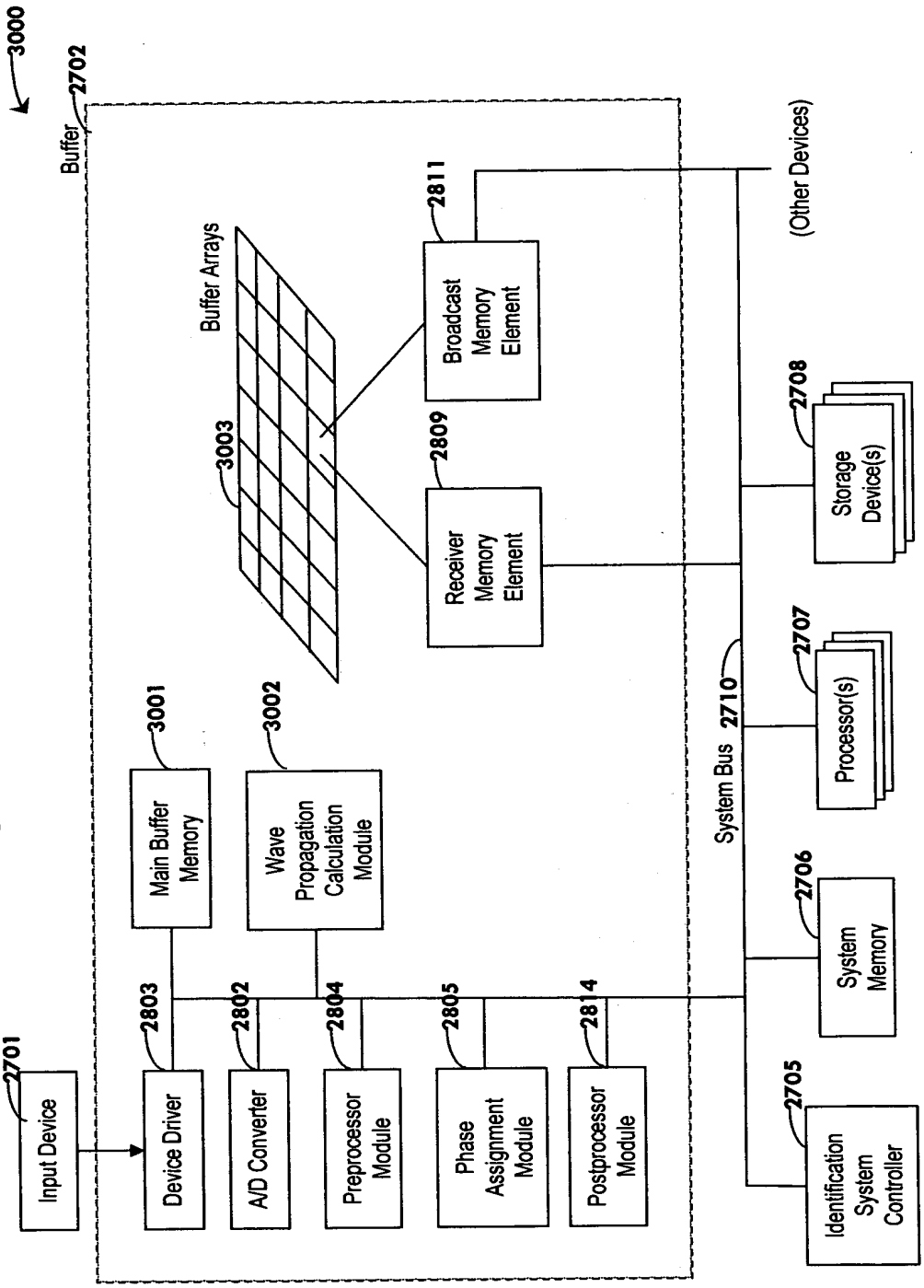

Buffer block diagram for software-based embodiment
3000
Buffer
2702
Input Device
2701
Device Driver
2803
A/D Converter
2802
Preprocessor Module
2804
Phase Assignment Module
2805
Postprocessor Module
2814
Main Buffer Memory
3001
Wave Propagation Calculation Module
3002
Buffer Arrays
3003
Receiver Memory Element
2809
Broadcast Memory Element
2811
System Bus
2710
Identification System Controller
2705
System Memory
2706
Processor(s)
2707
Storage Device(s)
2708
(Other Devices)

Figure 30

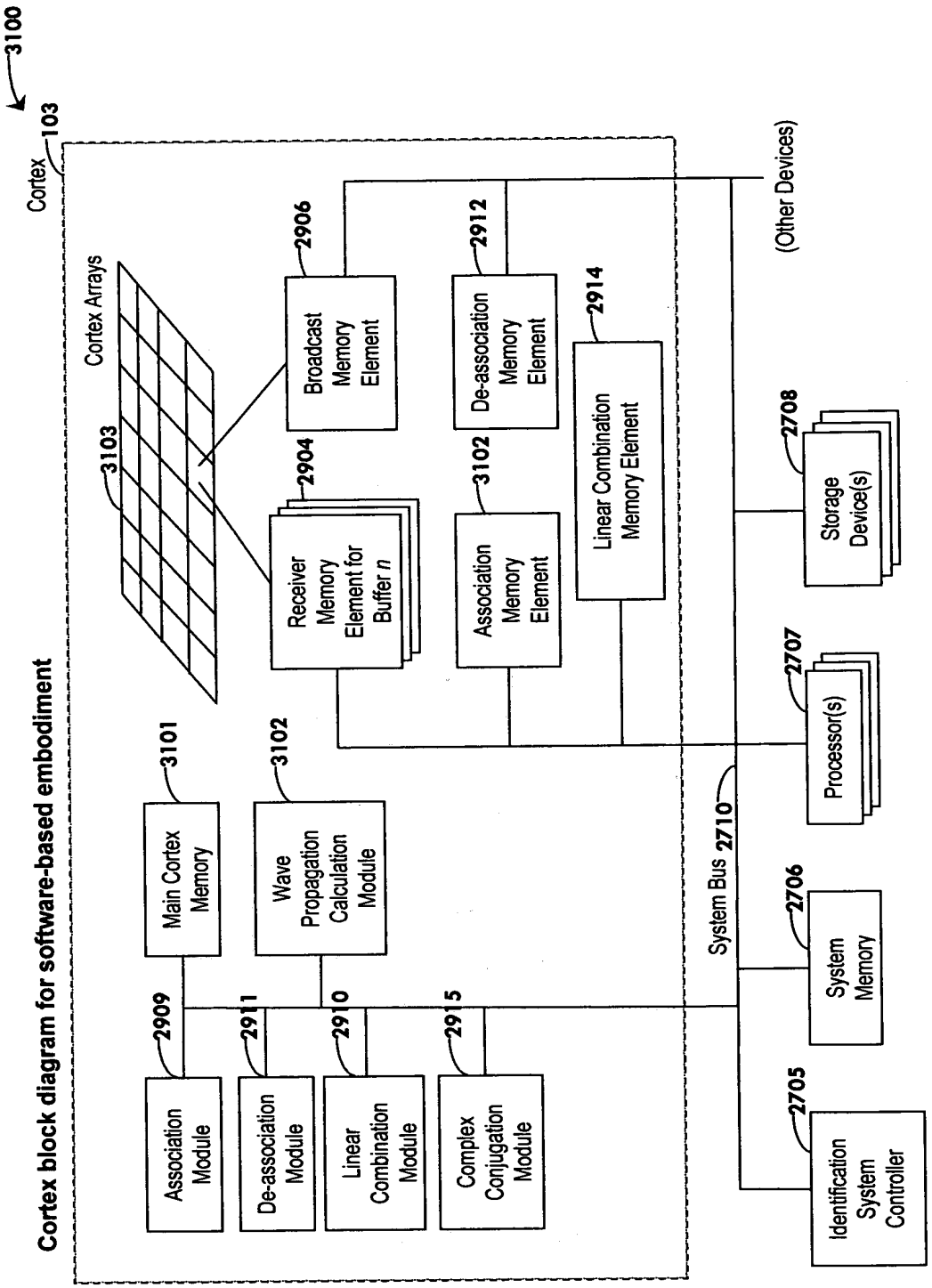

3100
Cortex 103
Cortex block diagram for software-based embodiment
Cortex Arrays
3103
Broadcast Memory Element
2906
Receiver Memory Element for Buffer n
2904
De-association Memory Element
2912
Association Memory Element
3102
Linear Combination Memory Element
2914
Main Cortex Memory
3101
Wave Propagation Calculation Module
3102
Association Module
2909
De-association Module
2911
Linear Combination Module
2910
Complex Conjugation Module
2915
System Bus 2710
Processor(s)
2707
Storage Device(s)
2708
System Memory
2706
Identification System Controller
2705
(Other Devices)

Figure 31

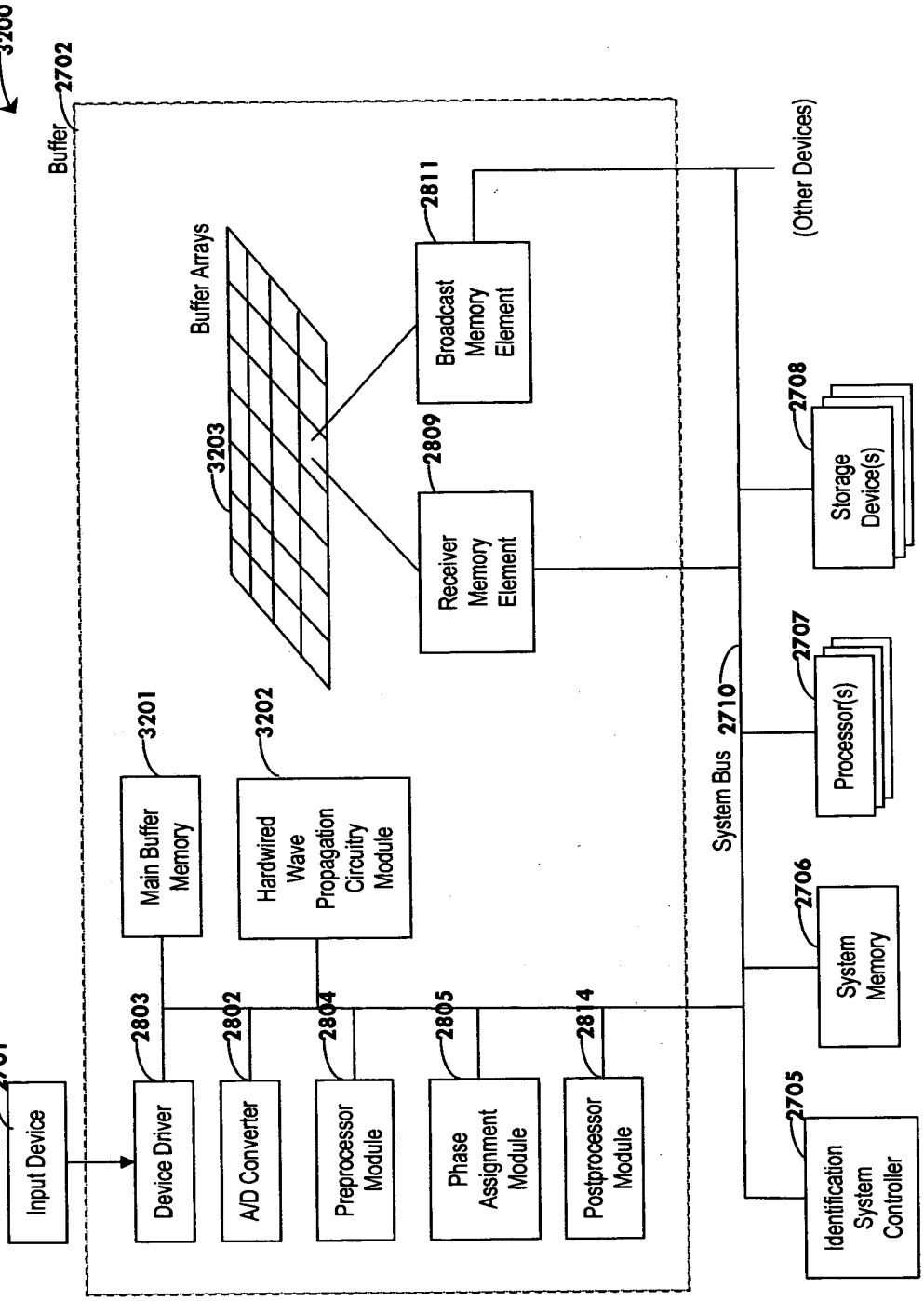

Buffer block diagram for hardwired circuitry-based embodiment
3200
Buffer
2702
Input Device
2701
Device Driver
2803
A/D Converter
2802
Preprocessor Module
2804
Phase Assignment Module
2805
Postprocessor Module
2814
Main Buffer Memory
3201
Hardwired Wave Propagation Circuitry Module
3202
Buffer Arrays
3203
Receiver Memory Element
2809
Broadcast Memory Element
2811
System Bus
2710
Identification System Controller
2705
System Memory
2706
Processor(s)
2707
Storage Device(s)
2708
(Other Devices)

Figure 32

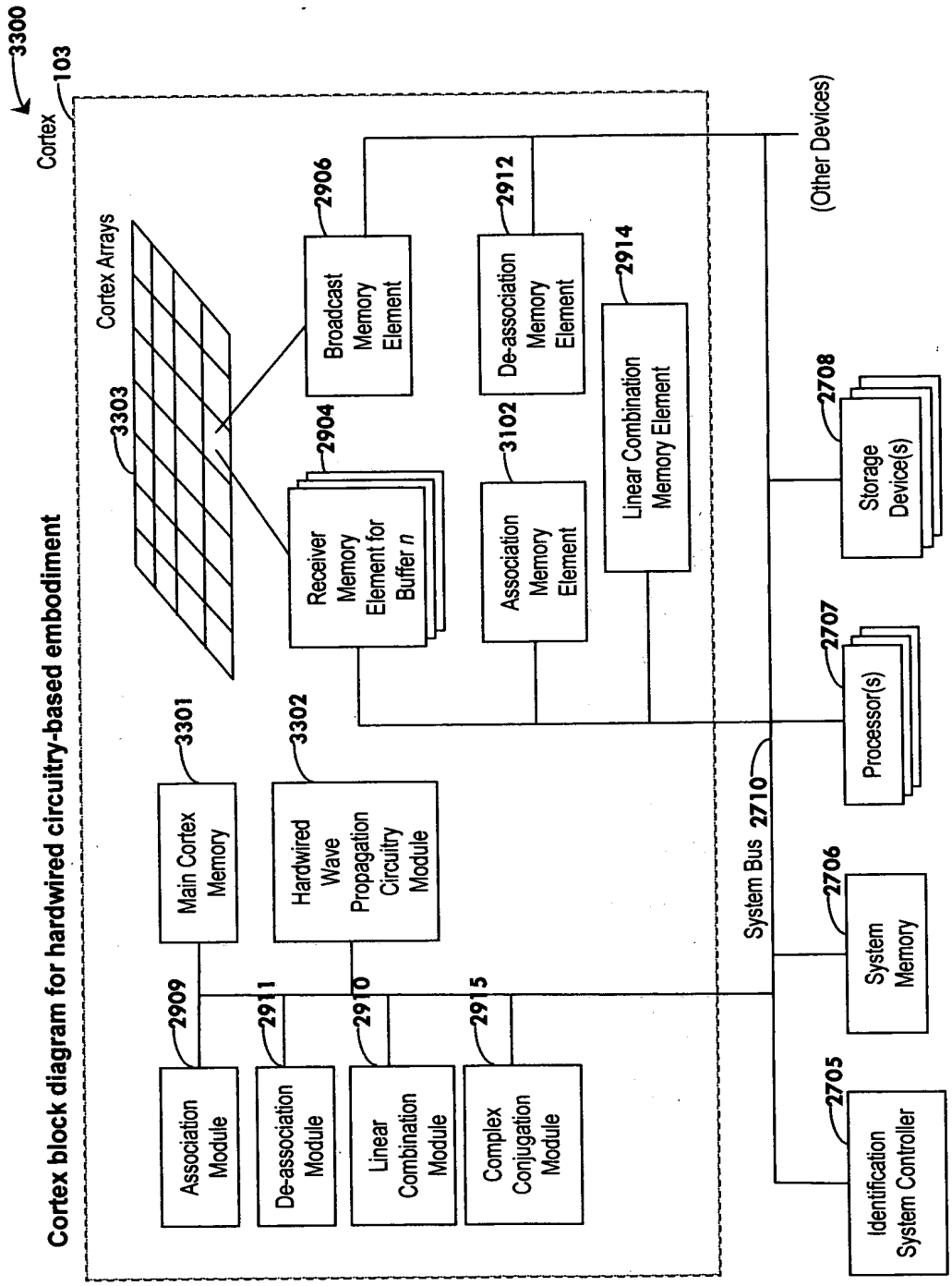

3300
Cortex
103
Cortex block diagram for hardwired circuitry-based embodiment
Cortex Arrays
3303
Broadcast Memory Element
2906
Receiver Memory Element for Buffer n
2904
De-association Memory Element
2912
Association Memory Element
3102
Linear Combination Memory Element
2914
Main Cortex Memory
3301
Hardwired Wave Propagation Circuitry Module
3302
Association Module
2909
De-association Module
2911
Linear Combination Module
2910
Complex Conjugation Module
2915
System Bus
2710
Processor(s)
2707
Storage Device(s)
2708
System Memory
2706
Identification System Controller
2705
(Other Devices)

Figure 33

Figure 34

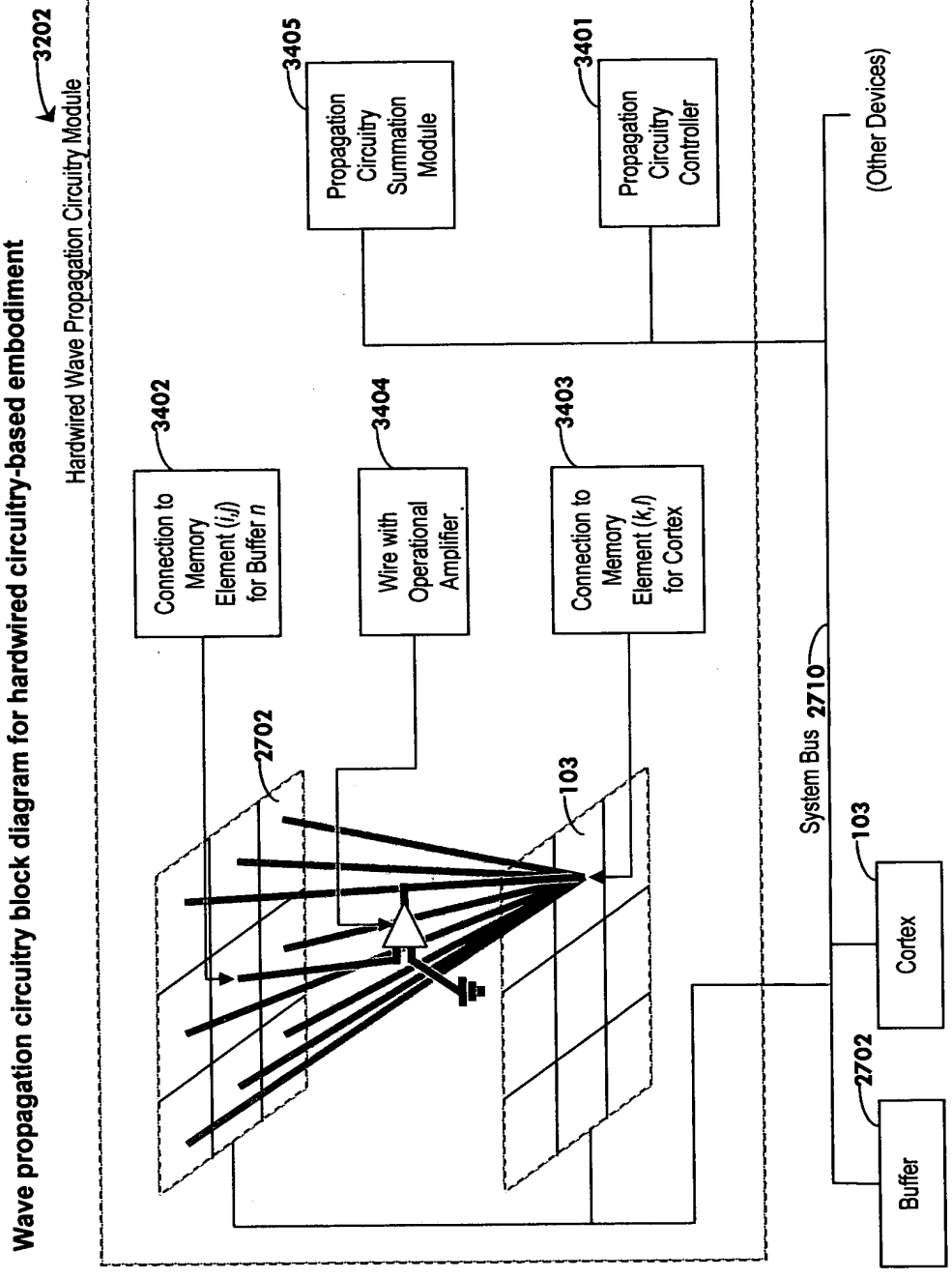

Wave propagation circuitry block diagram for hardwired circuitry-based embodiment
3202
Hardwired Wave Propagation Circuitry Module
3405
Propagation Circuitry Summation Module
3401
Propagation Circuitry Controller
3402
Connection to Memory Element (i,j) for Buffer n
3404
Wire with Operational Amplifier
3403
Connection to Memory Element (k,l) for Cortex
2702
103
System Bus 2710
(Other Devices)
103
Cortex
2702
Buffer

Figure 35

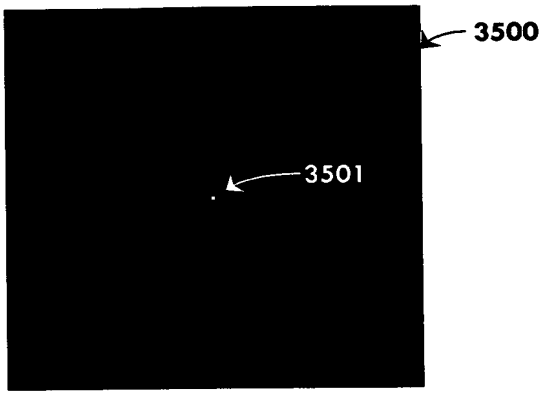


Figure 36

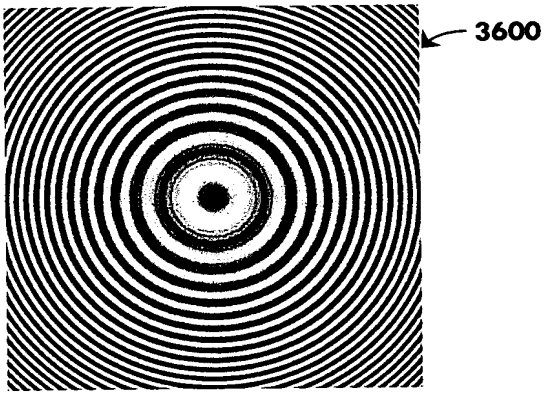


Figure 37

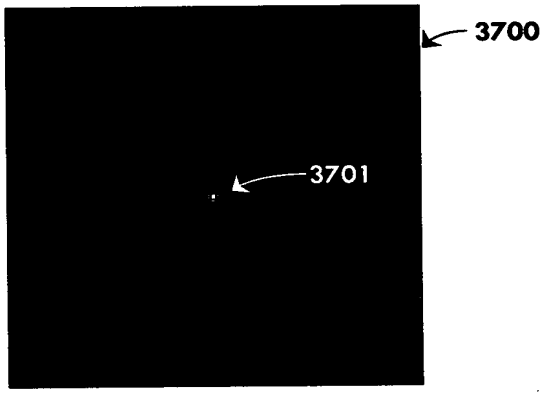


Figure 38

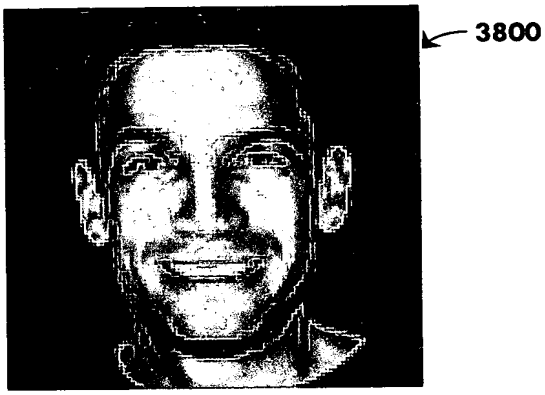


Figure 39

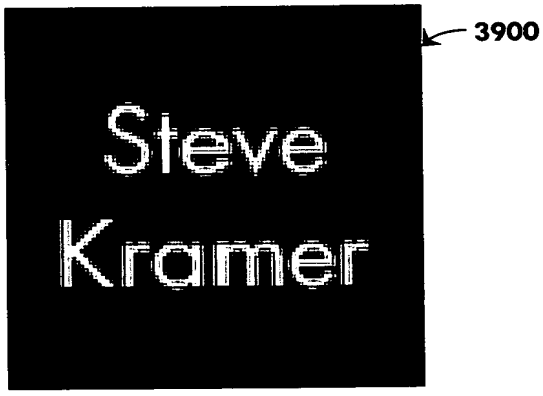


Figure 40

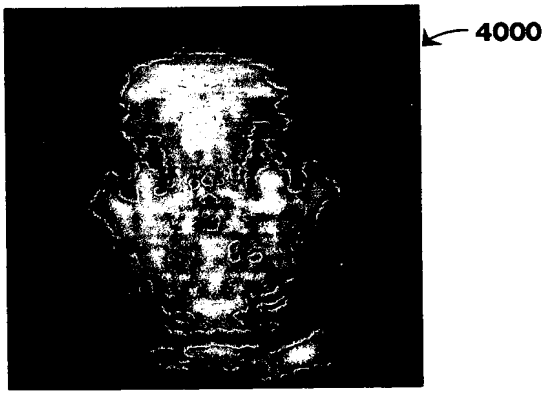


Figure 41

Figure 42

Figure 43

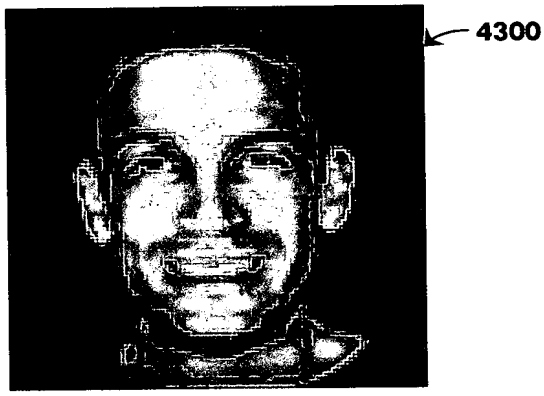


Figure 44

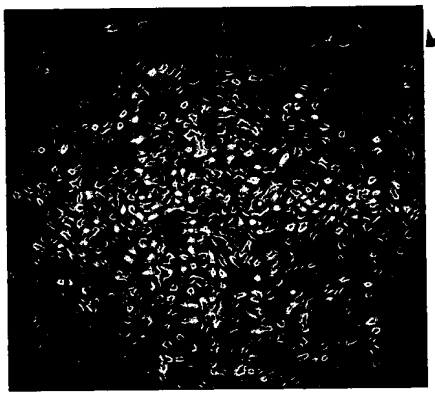


Figure 45

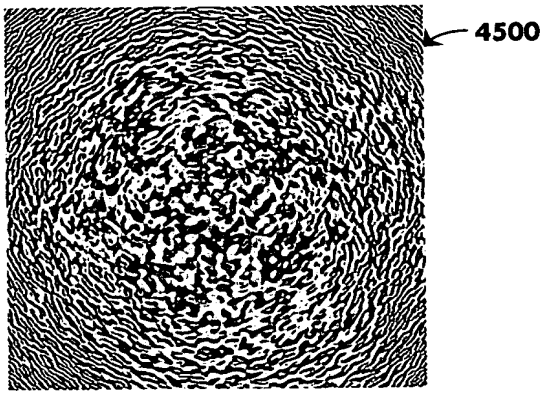


Figure 46

Figure 47

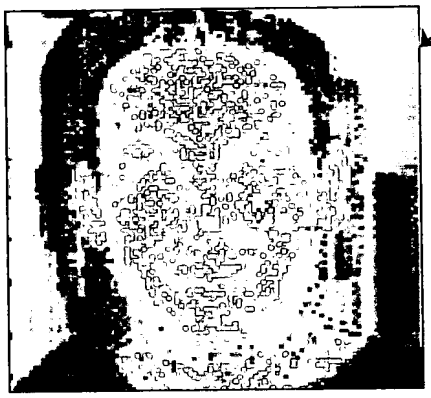


Figure 48A

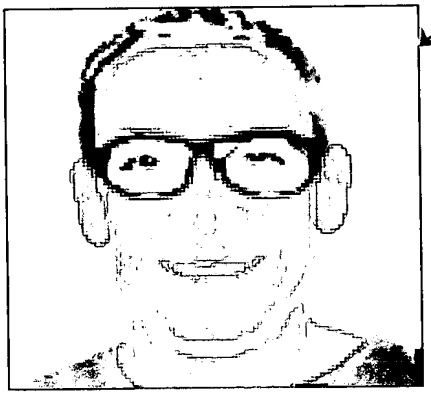


Figure 48B

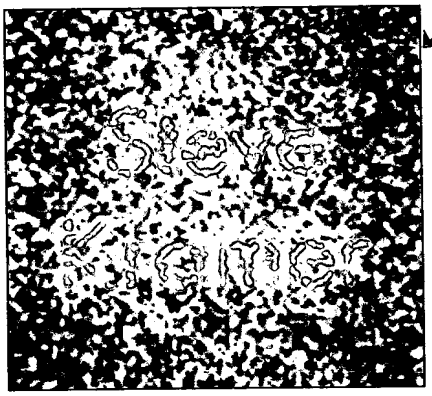


Figure 49A

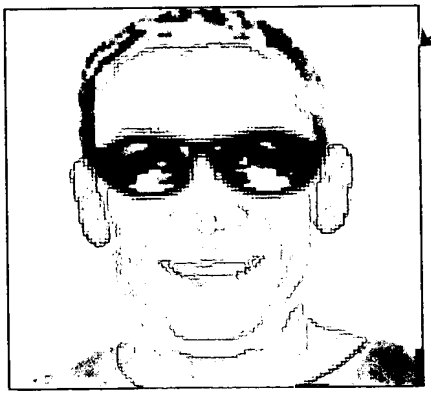


Figure 49B

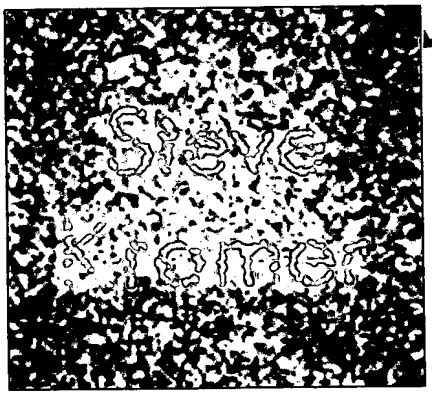


Figure 50A

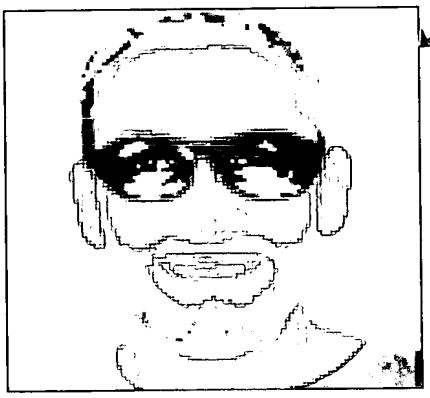


Figure 50B

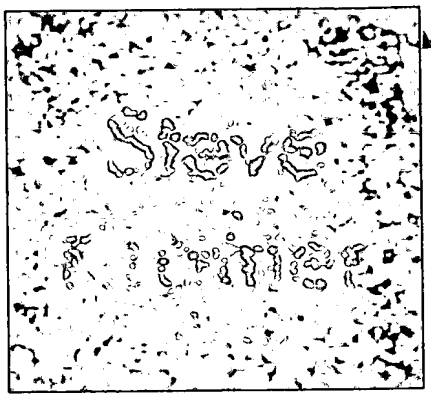


Figure 51A

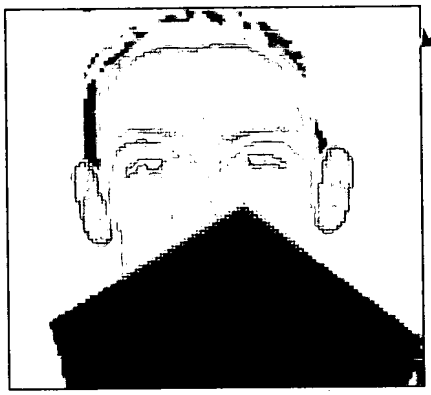


Figure 51B

Figure 52A

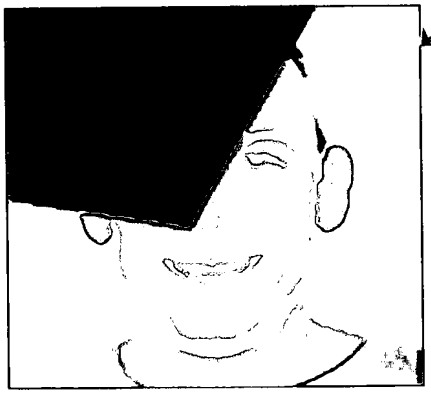


Figure 52B

Figure 53A

Figure 53B

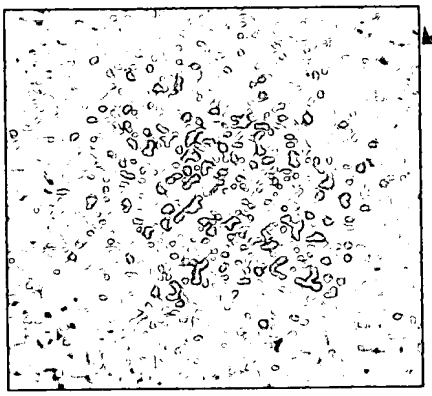


Figure 54A

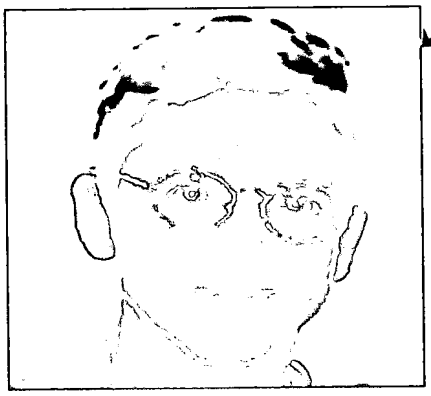


Figure 54B

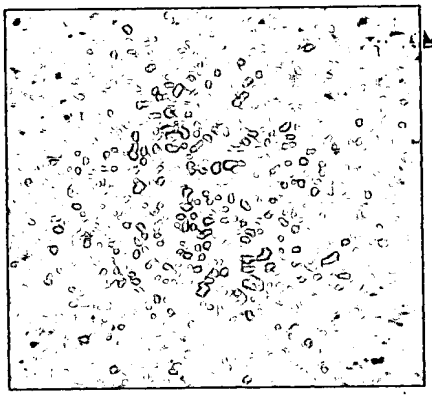


Figure 55A

Figure 55B

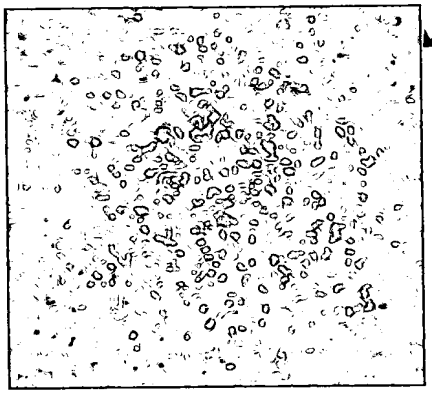


Figure 56

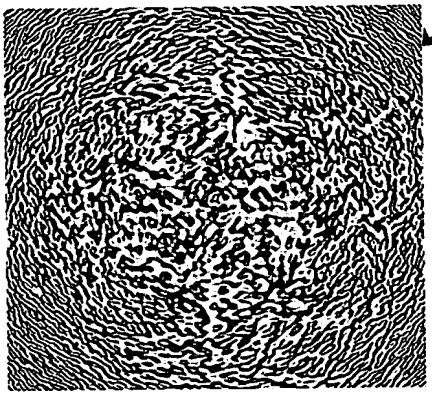


Figure 57

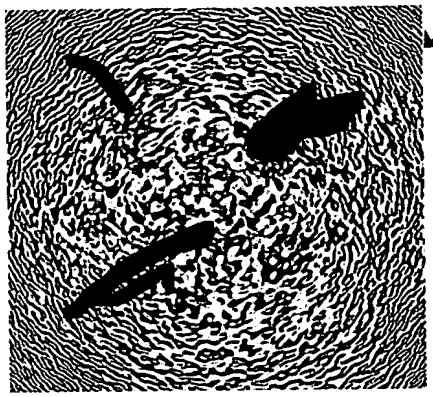


Figure 58A

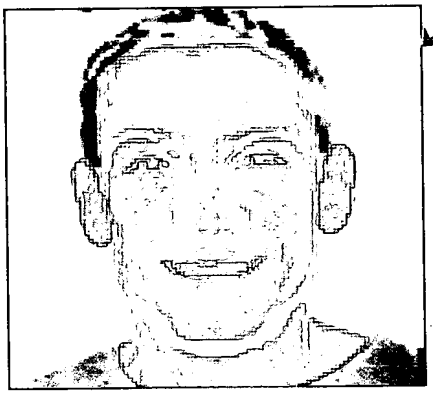


Figure 58B

Figure 59A

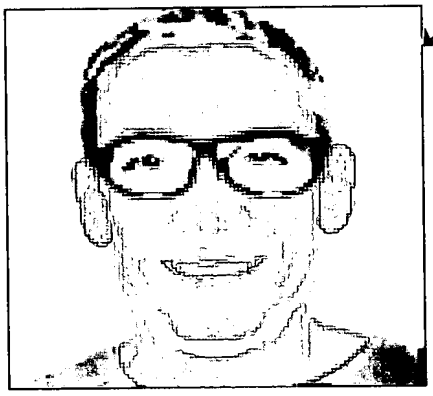


Figure 59B

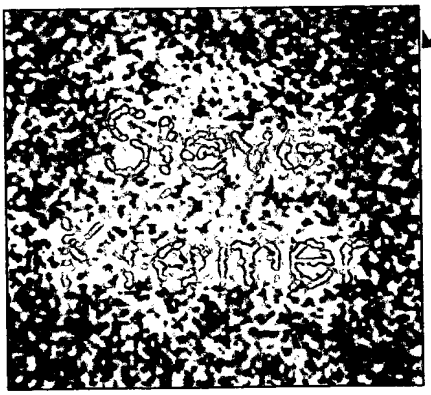


Figure 60A

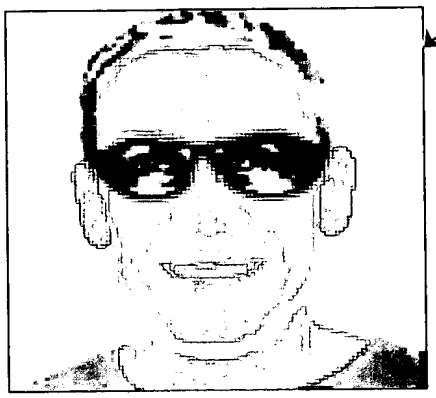


Figure 60B

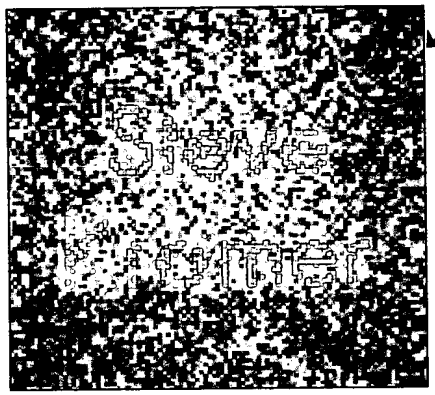


Figure 61A

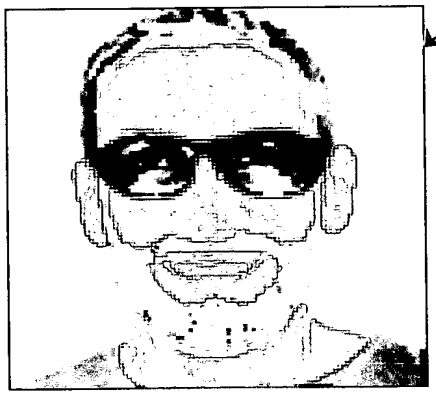


Figure 61B

Figure 62A

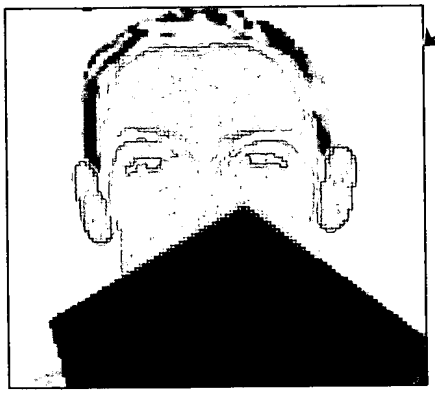


Figure 62B

Figure 63A

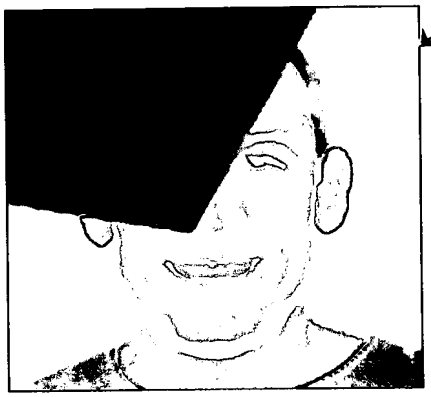


Figure 63B

Figure 64

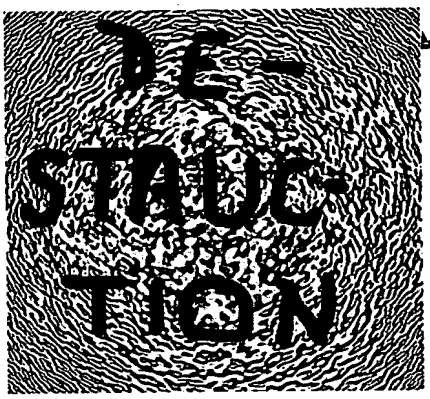


Figure 65A

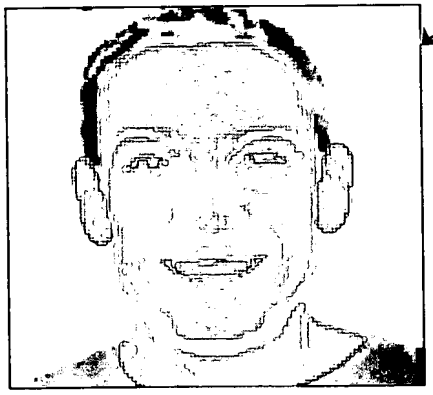


Figure 65B

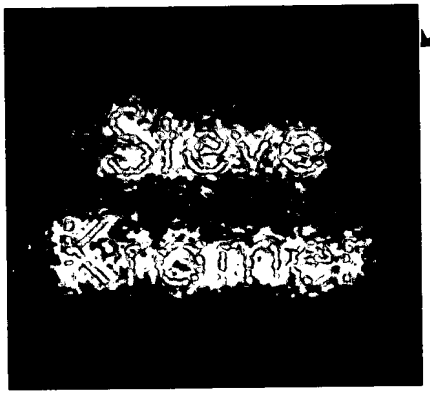


Figure 66A

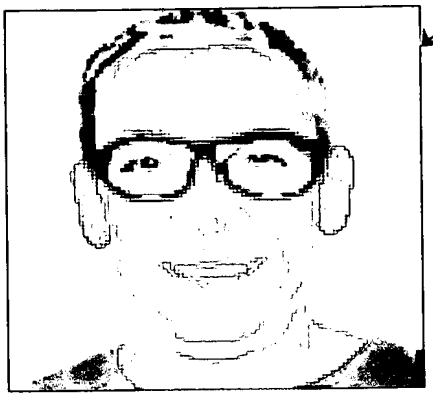


Figure 66B

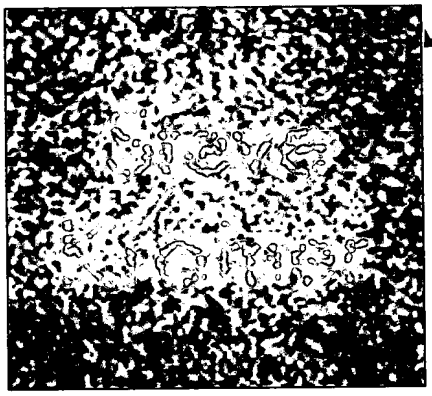


Figure 67A

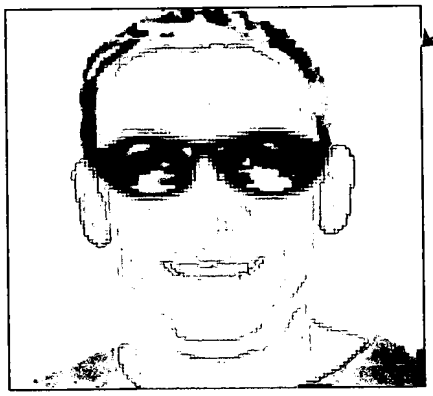


Figure 67B

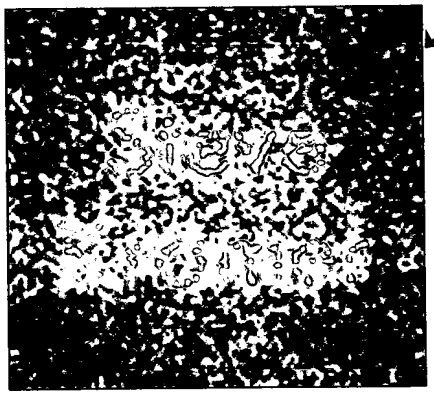


Figure 68A

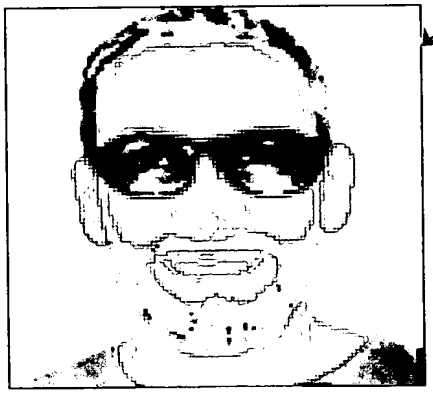


Figure 68B

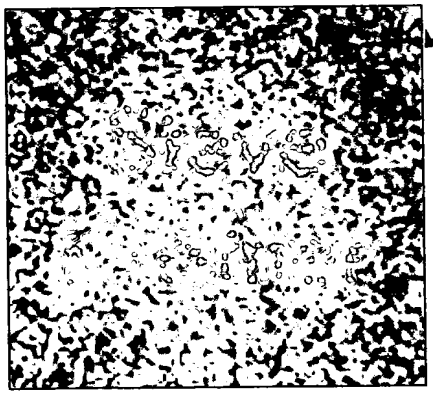


Figure 69A

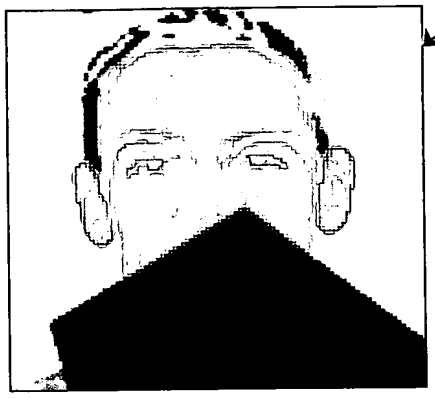


Figure 69B

Figure 70A

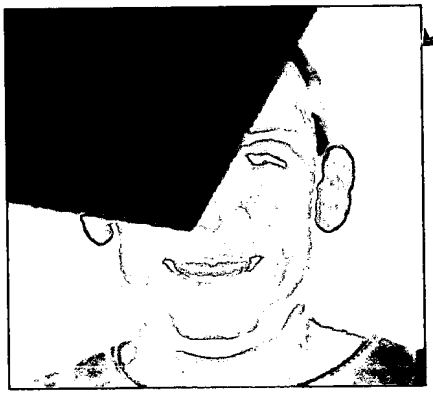


Figure 70B

Figure 71

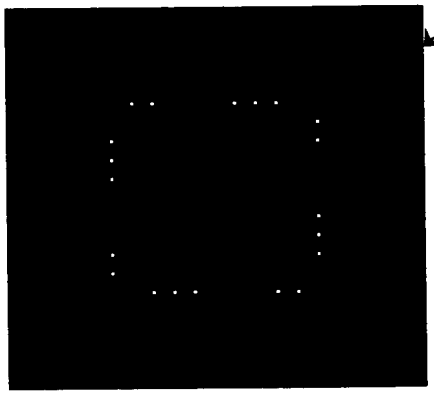


Figure 72

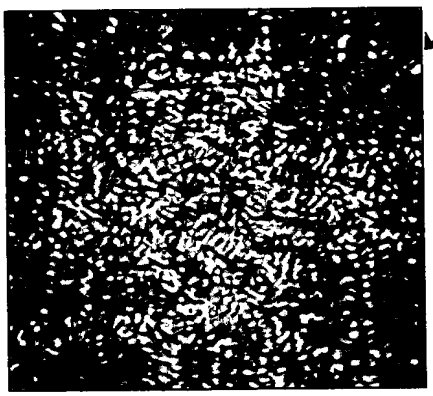


Figure 73

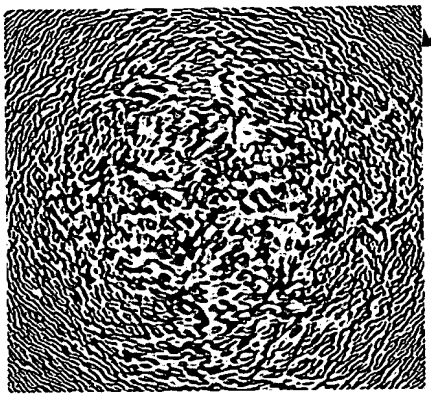


Figure 74A

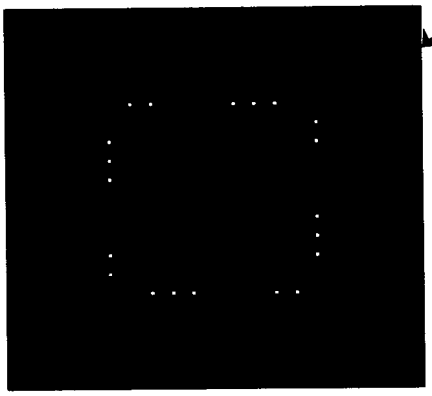


Figure 74B

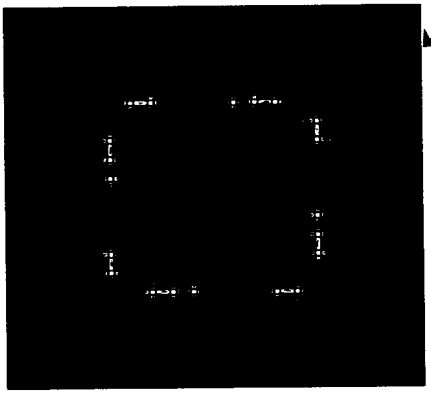


Figure 75A

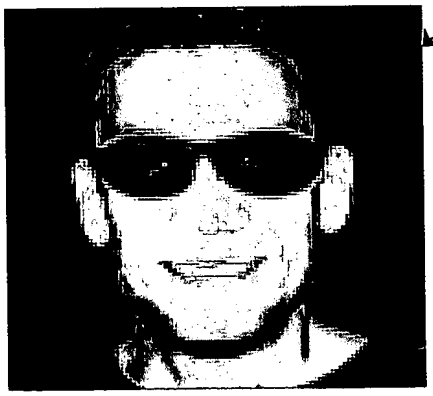


Figure 75B

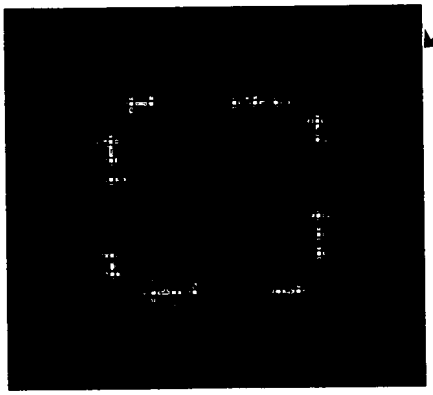


Figure 76A

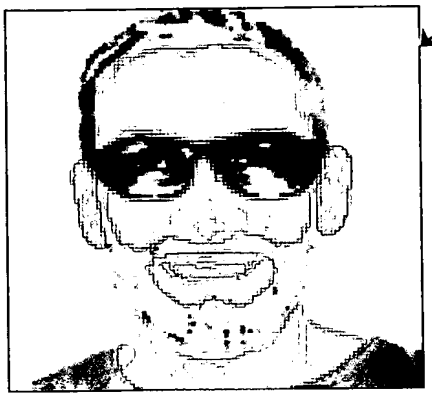


Figure 76B

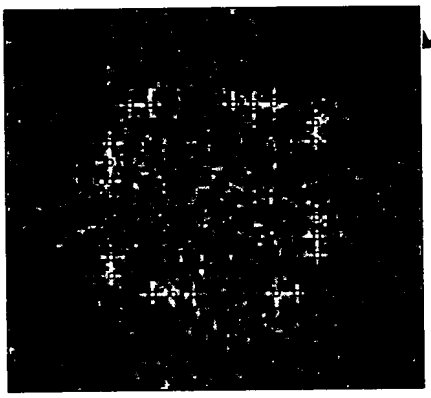


Figure 77A

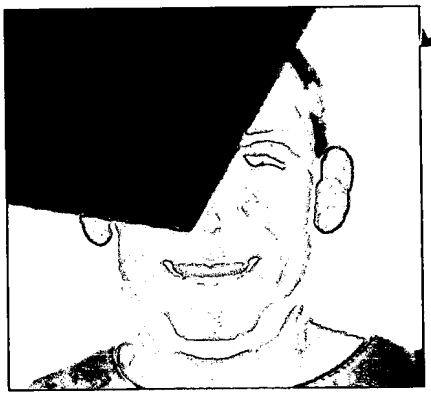


Figure 77B

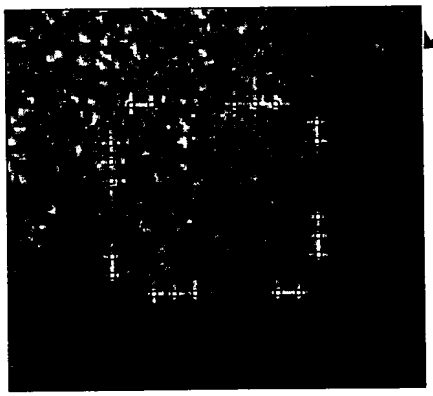


Figure 78A

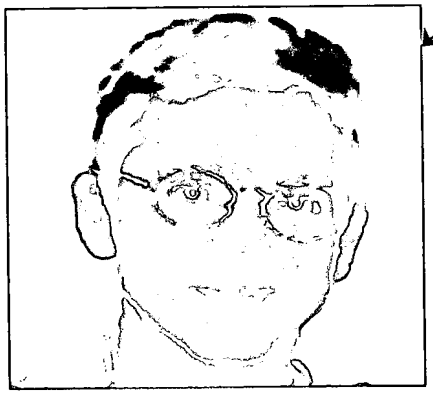


Figure 78B

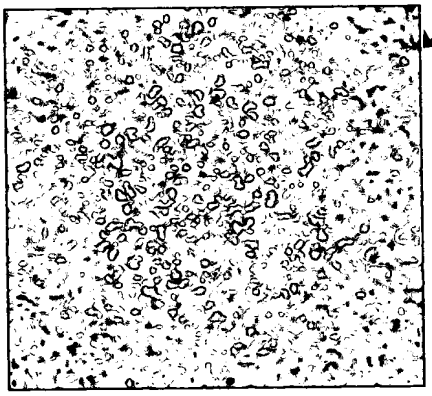


Figure 79

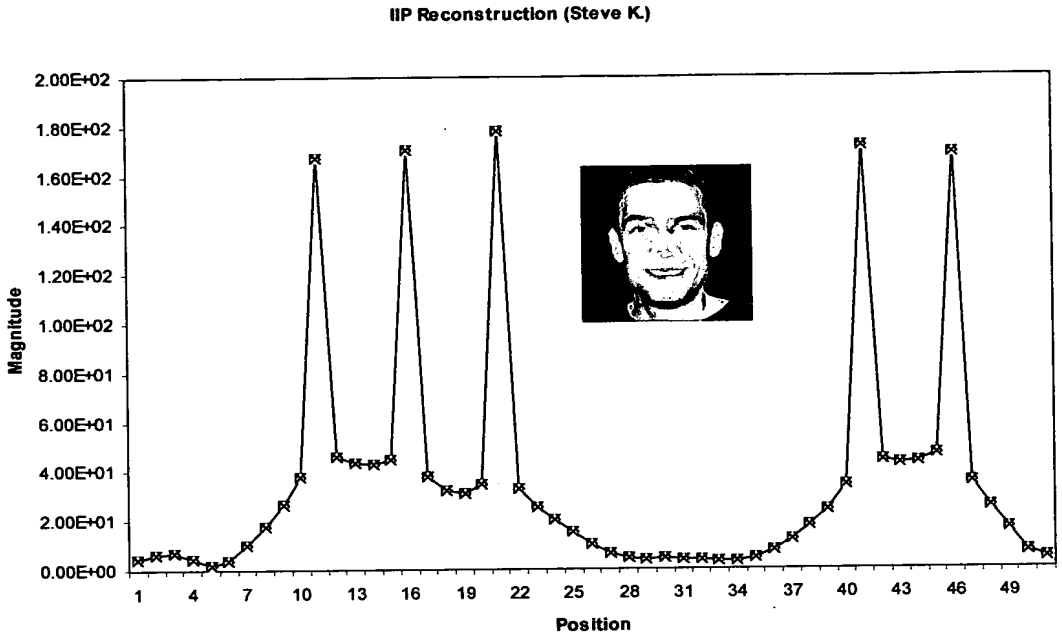


Figure 80

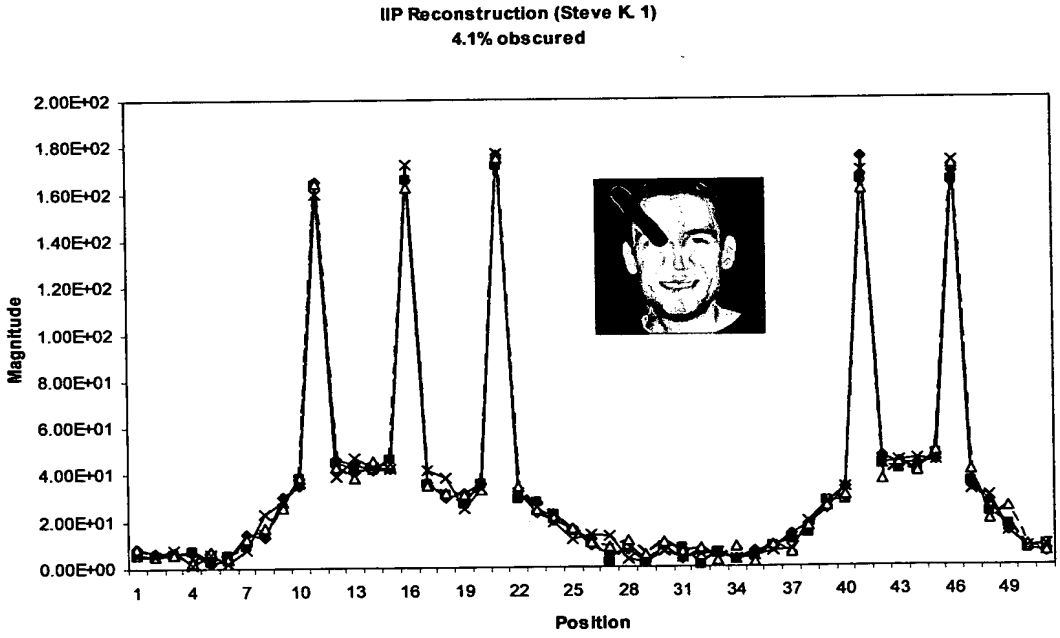


Figure 81

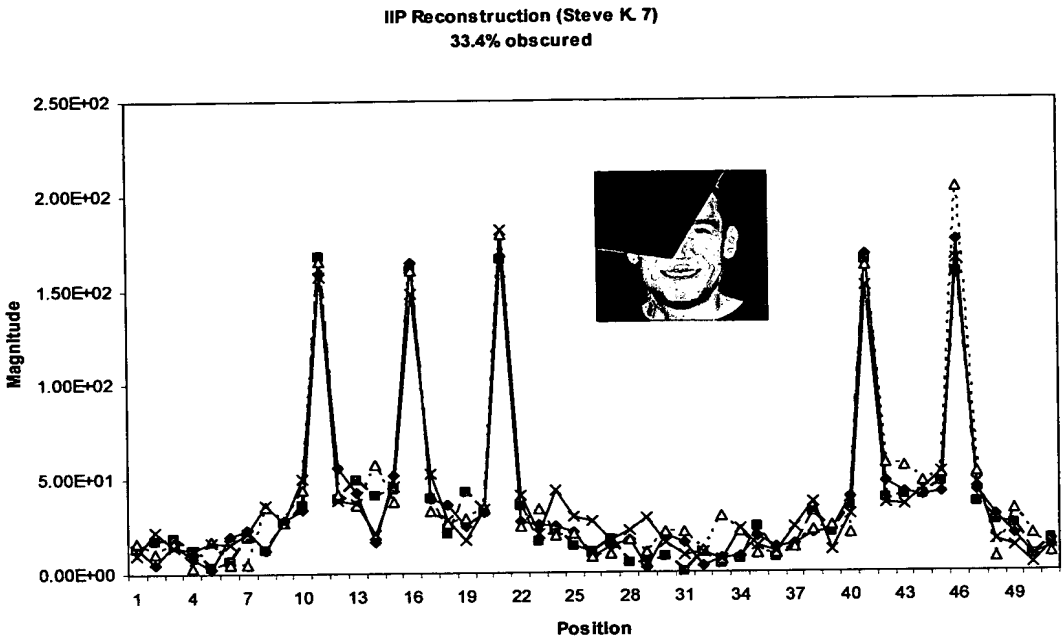


Figure 82

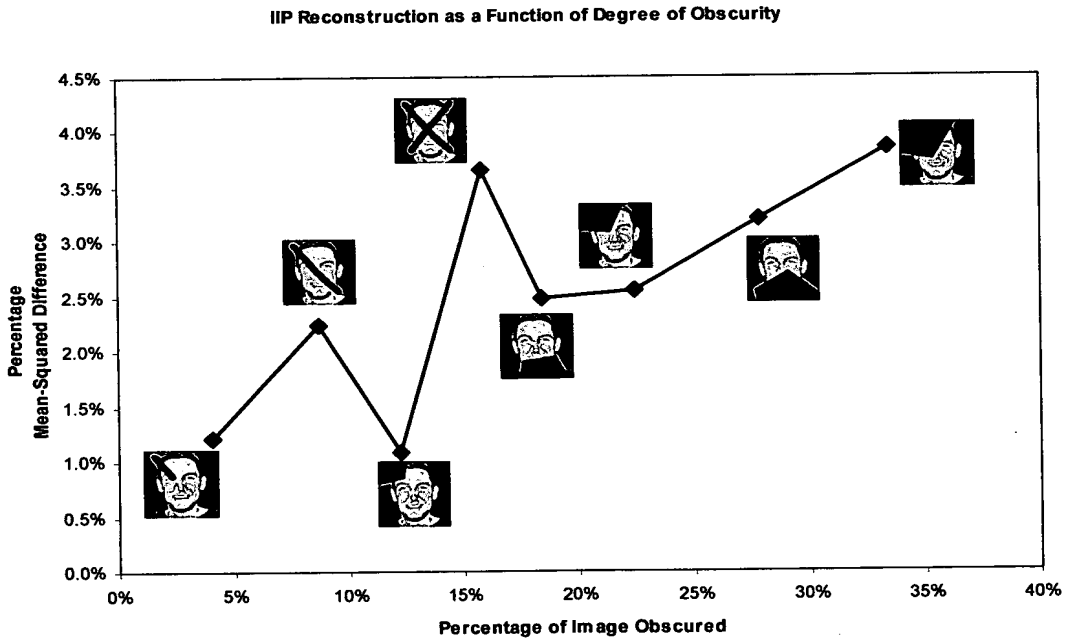


Figure 83A

Figure 83B

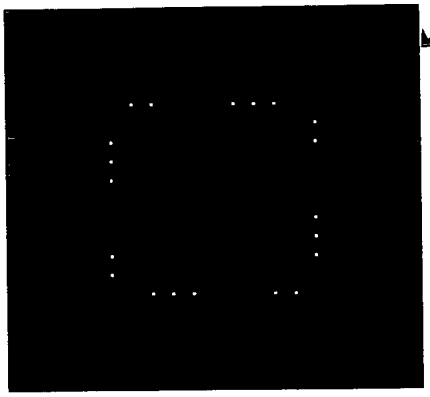


Figure 84A

Figure 84B

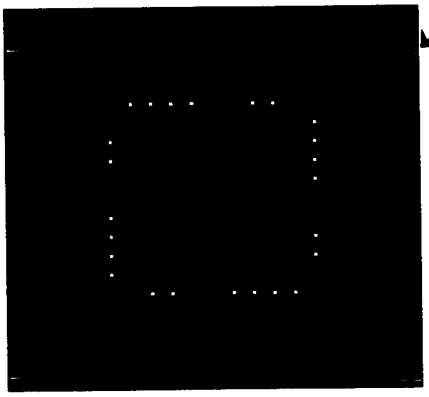


Figure 85A

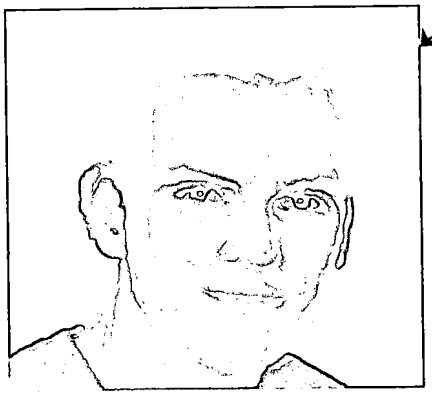


Figure 85B

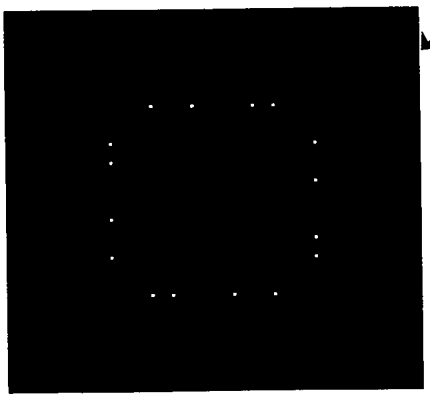


Figure 86A

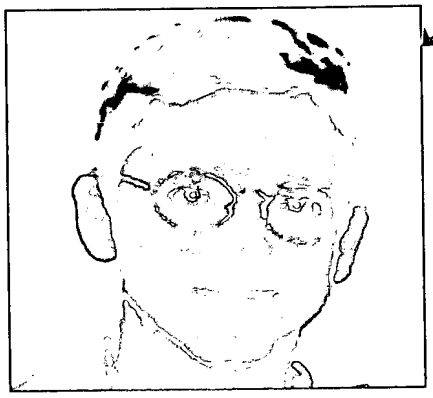


Figure 86B

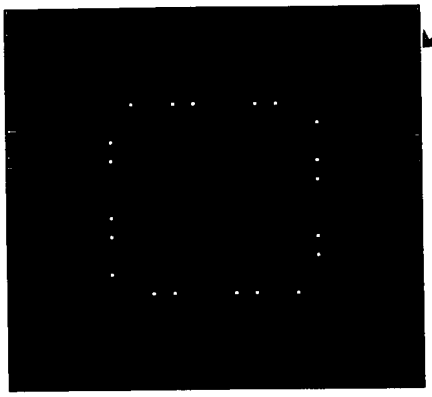


Figure 87A

Figure 87B

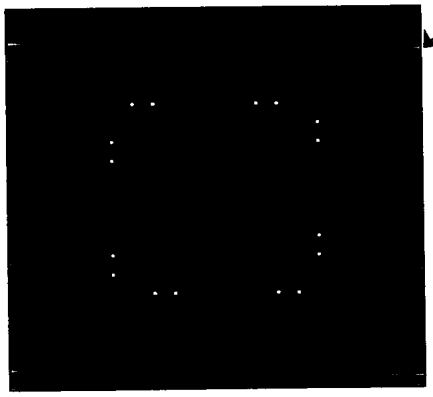


Figure 88A

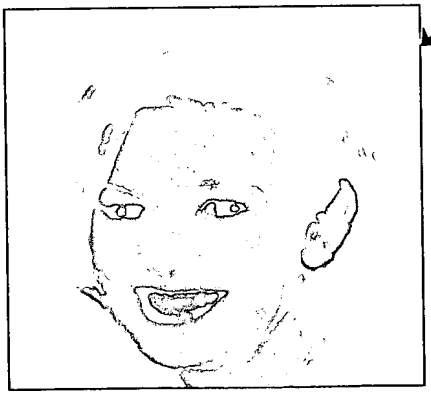


Figure 88B

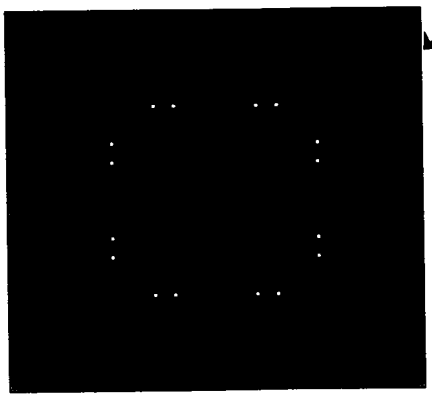


Figure 89A

Figure 89B

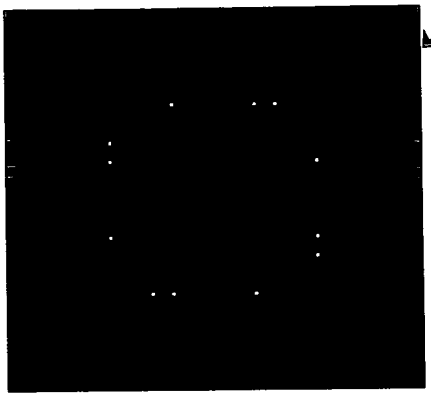


Figure 90

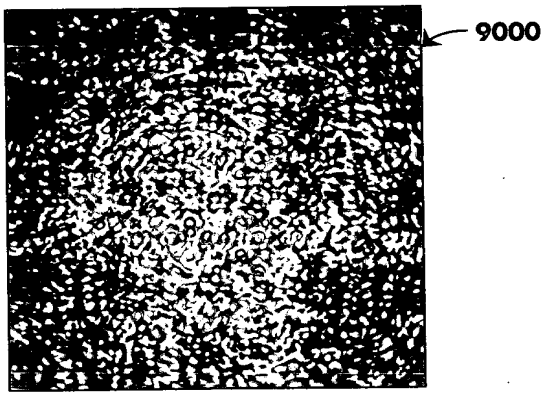


Figure 91

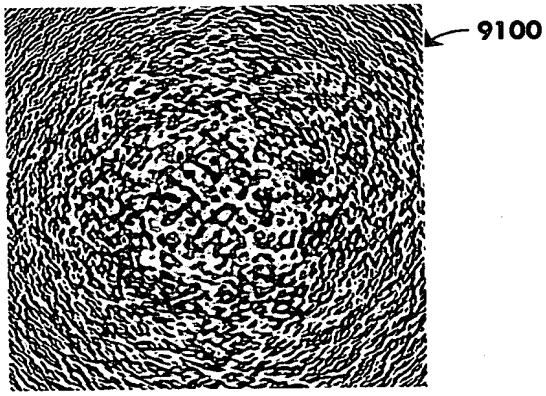


Figure 92A

Figure 92B

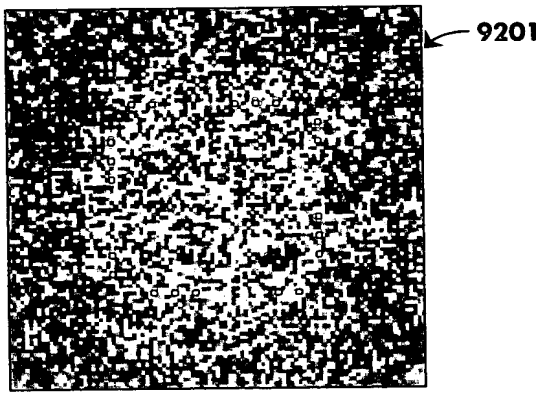


Figure 93A

Figure 93B

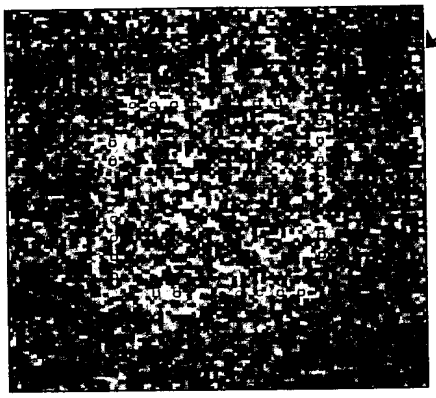


Figure 94A

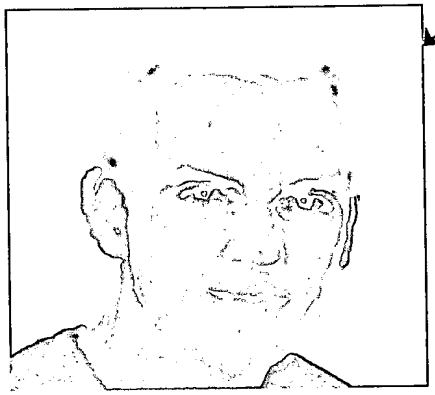


Figure 94B

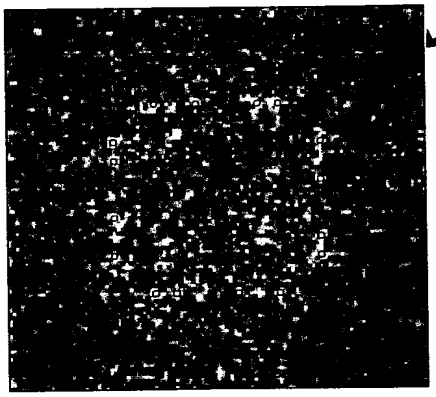


Figure 95A

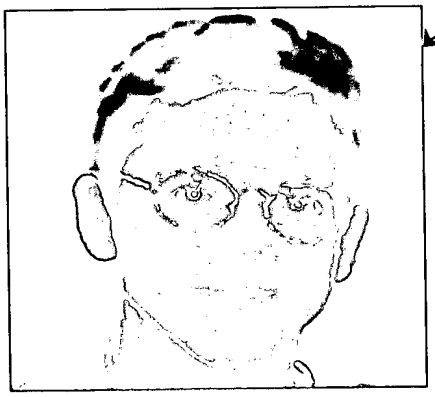


Figure 95B

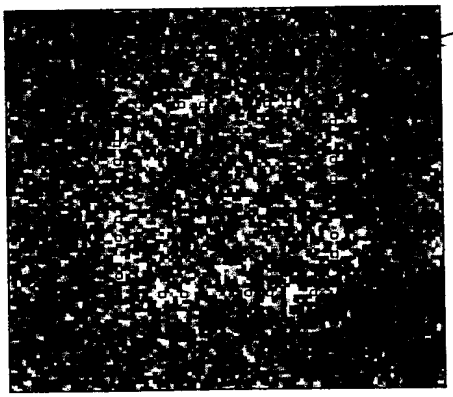


Figure 96A

Figure 96B

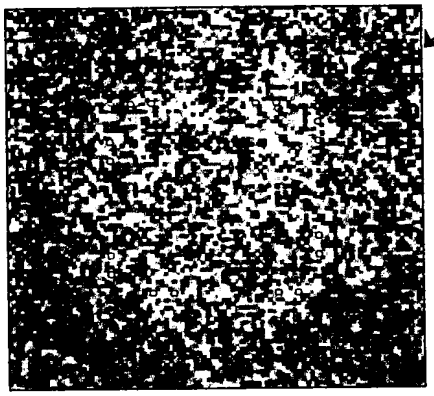


Figure 97A

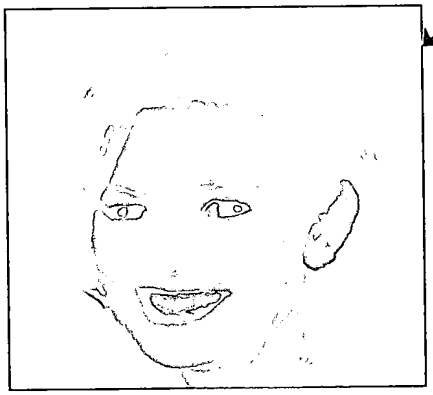


Figure 97B

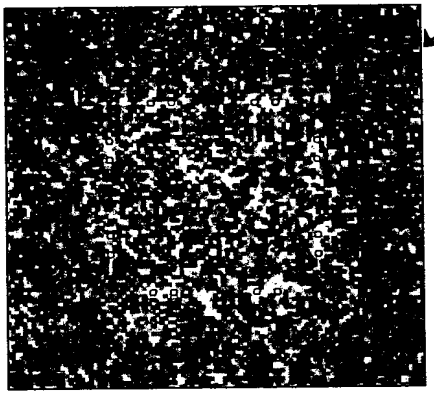


Figure 98A

Figure 98B

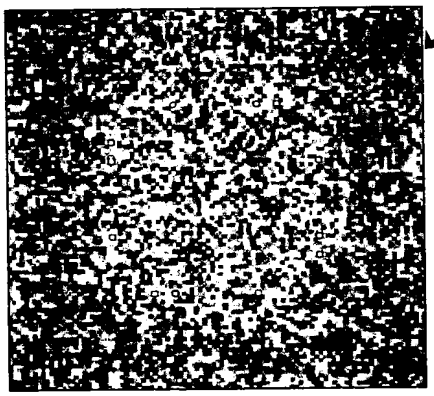


Figure 99A

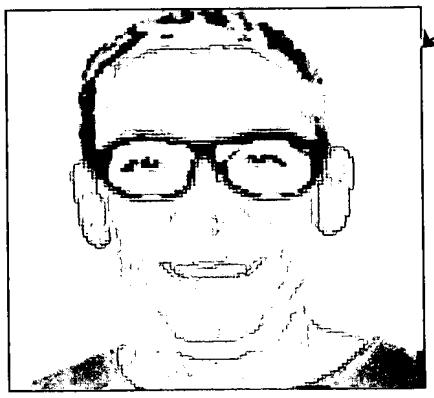


Figure 99B

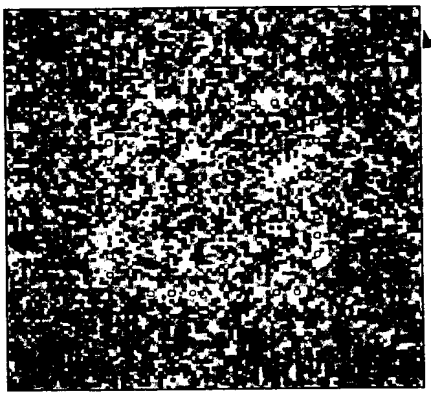


Figure 100A

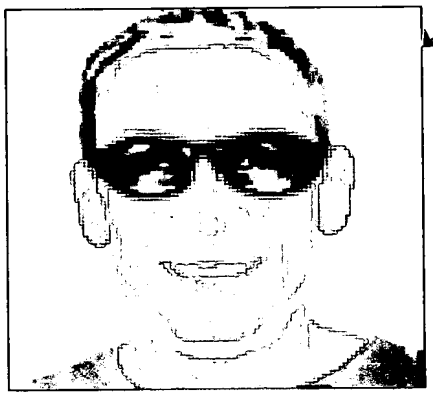


Figure 100B

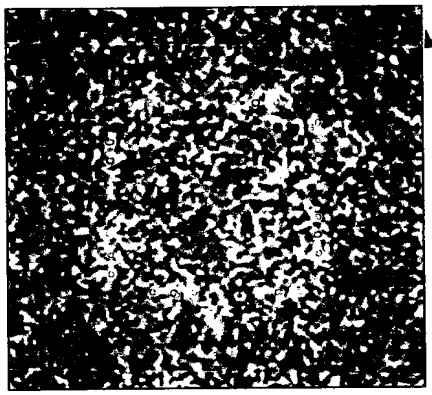


Figure 101A

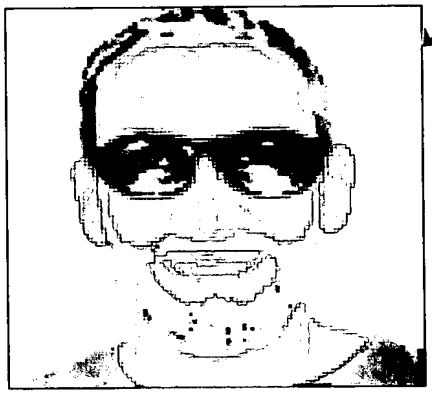


Figure 101B

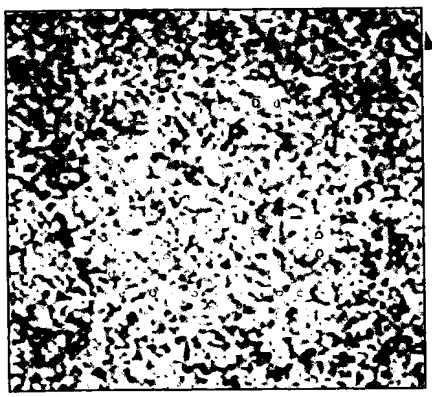


Figure 102A

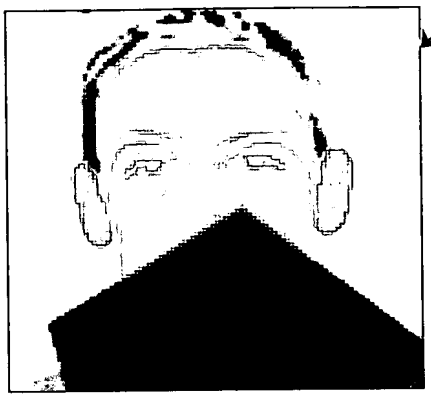


Figure 102B

Figure 103A

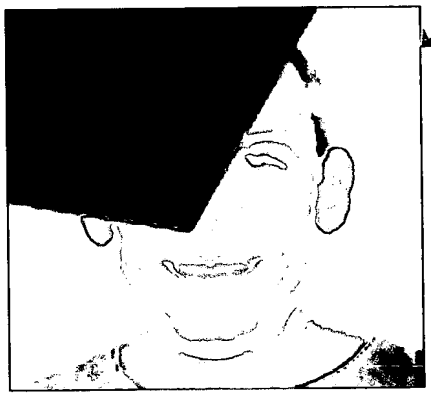


Figure 103B

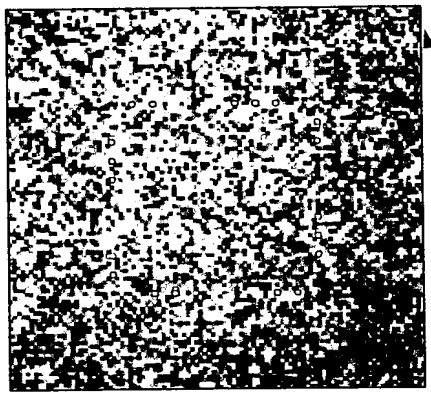


Figure 104A

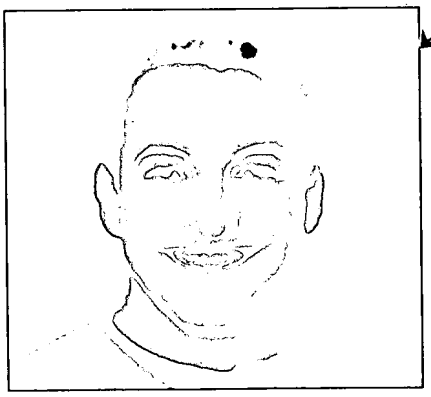


Figure 104B

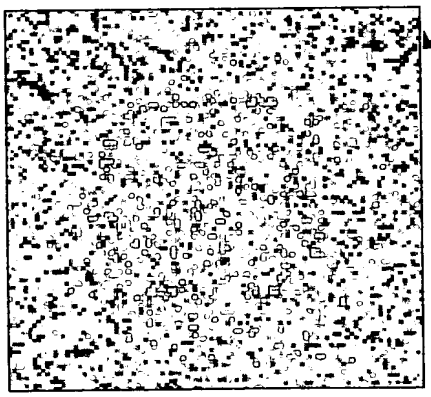


Figure 105

Figure 106

Figure 107

Figure 108

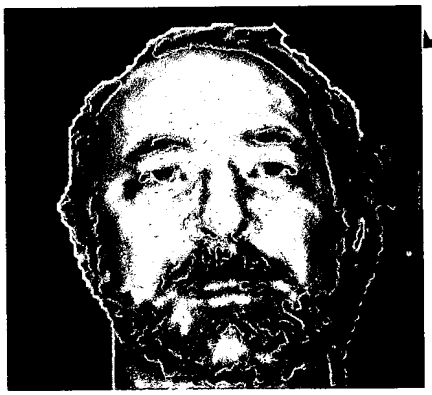


Figure 109

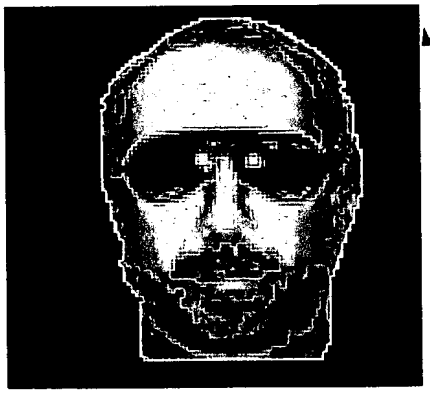


Figure 110

Figure 111

Figure 112

Figure 113

Figure 114

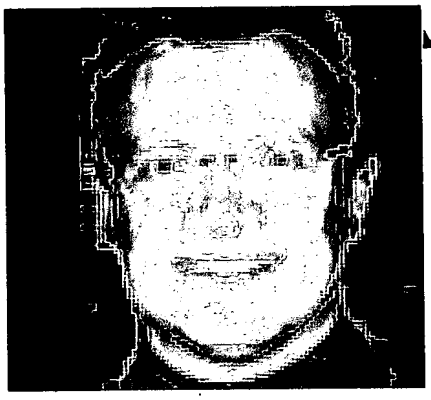


Figure 115

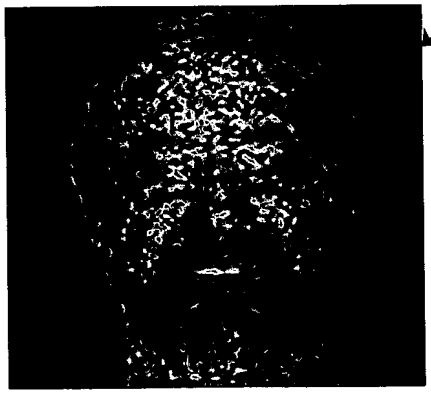


Figure 116

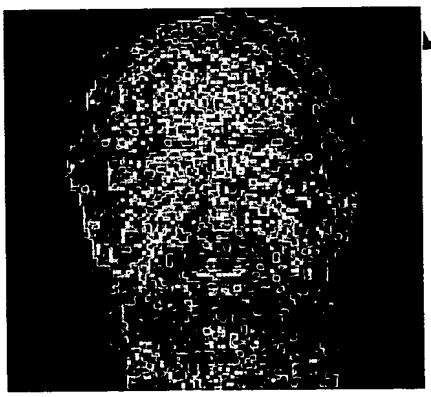


Figure 117

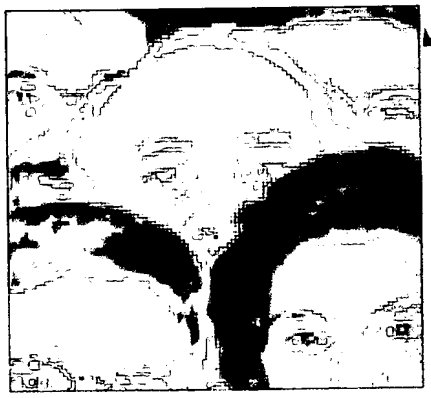


Figure 118A

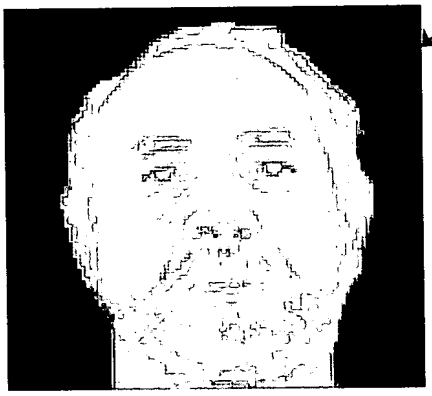


Figure 118B

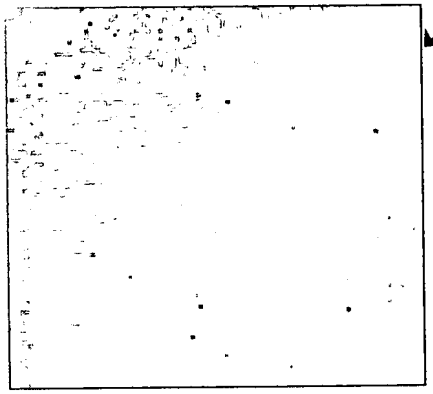


Figure 119A

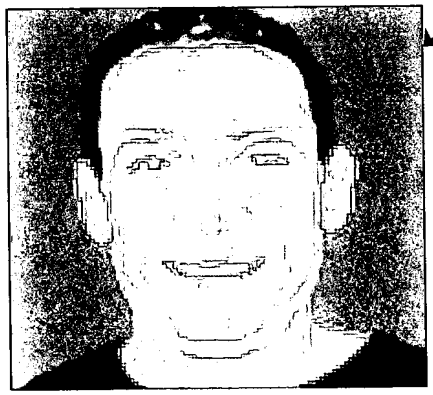


Figure 119B

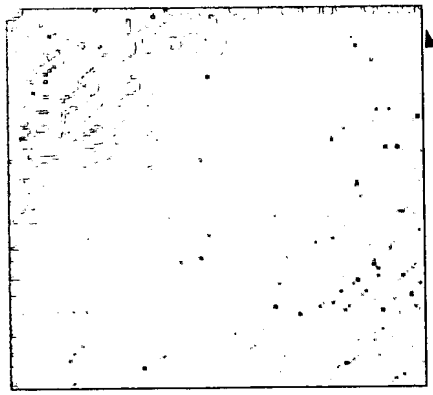


Figure 120A

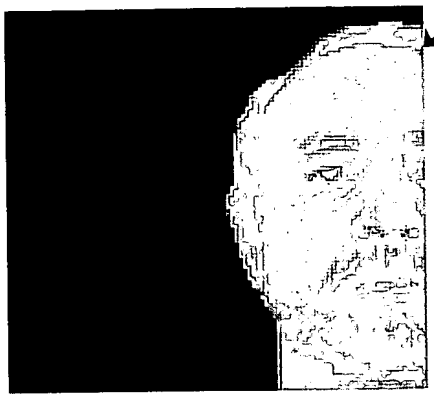


Figure 120B

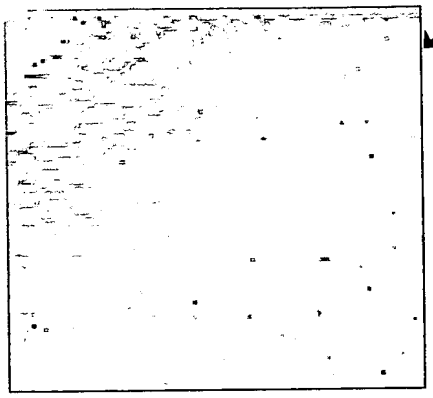


Figure 121

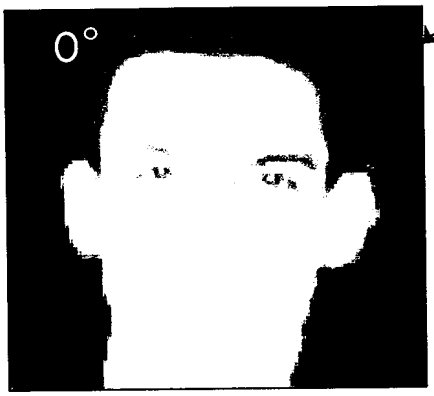


Figure 122

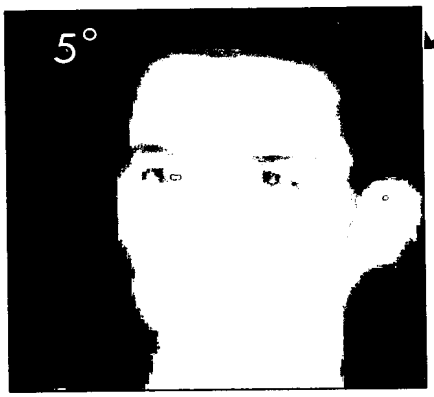


Figure 123

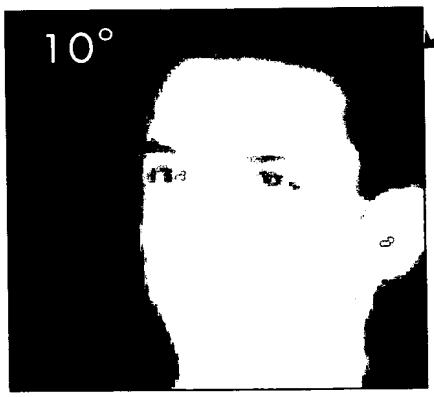


Figure 124

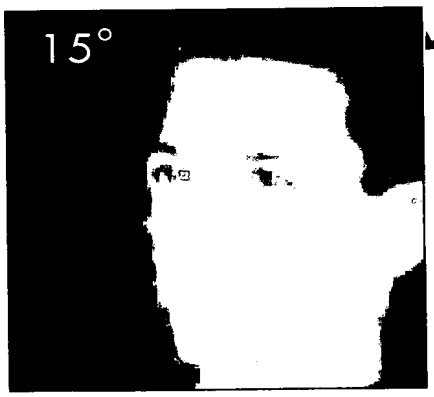


Figure 125

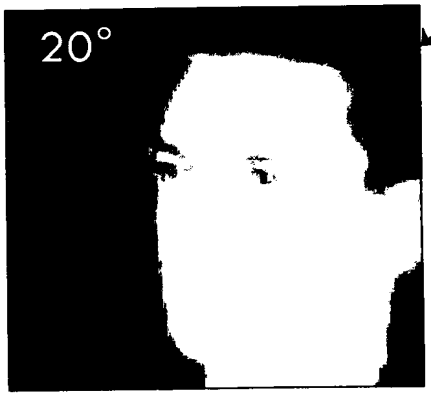


Figure 126

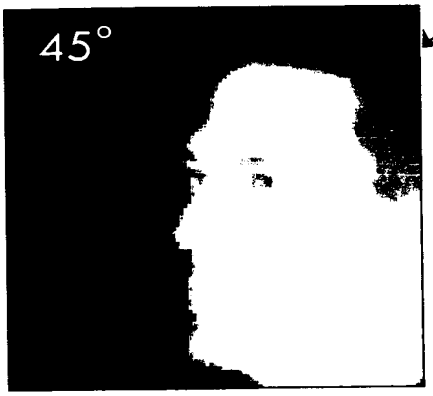


Figure 127

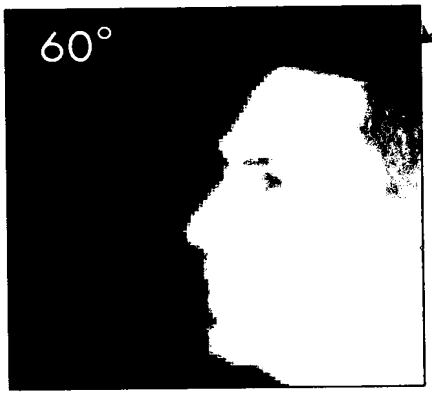


Figure 128

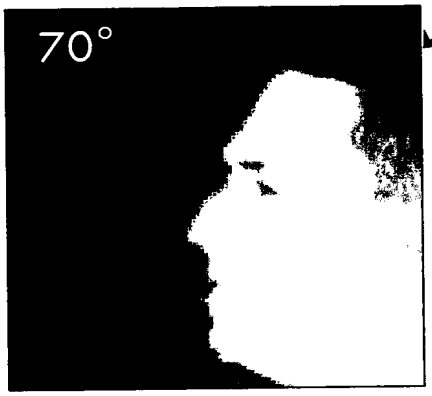


Figure 129

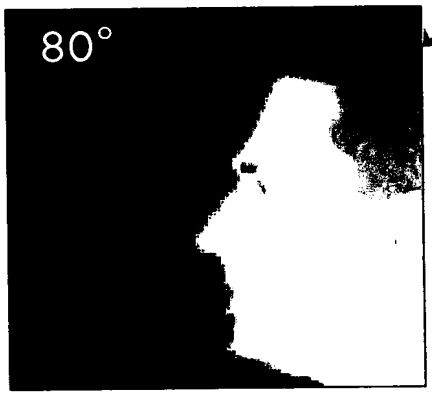


Figure 130

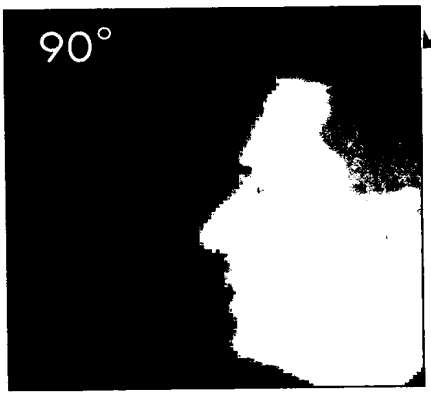


Figure 131

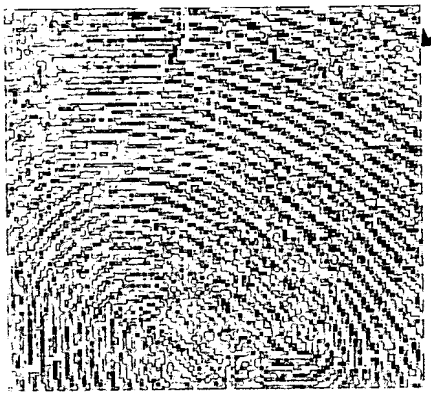


Figure 132

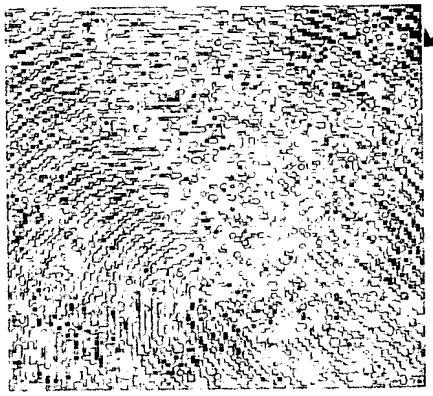


Figure 133

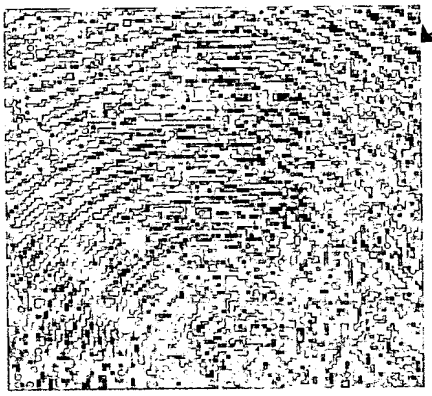


Figure 134

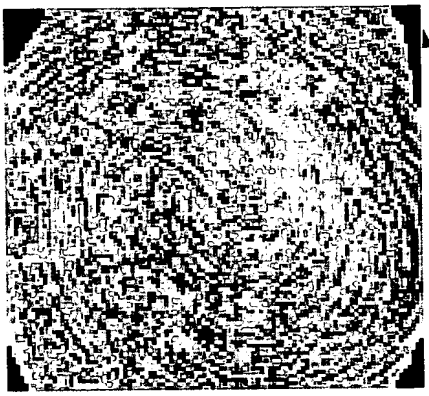


Figure 135

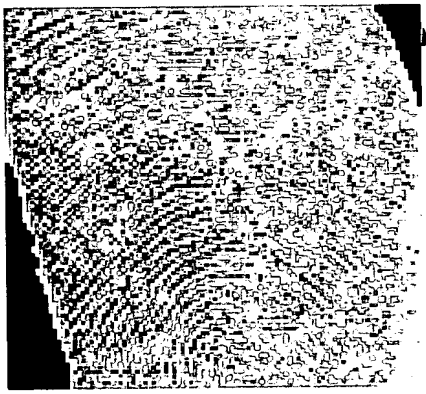


Figure 136

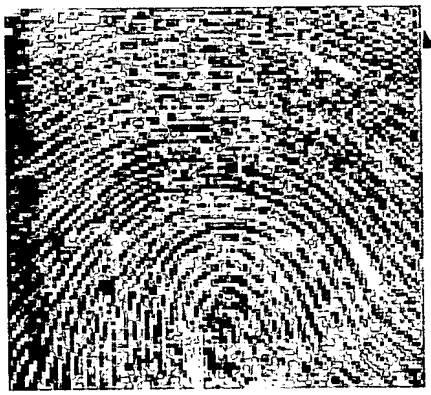


Figure 137

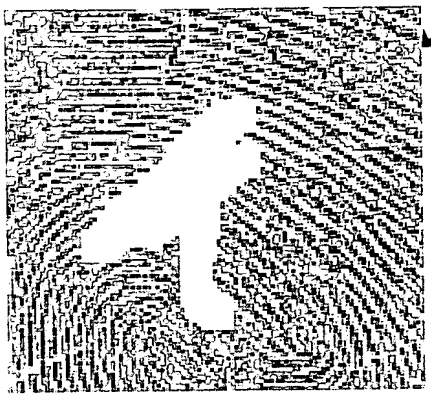


Figure 138

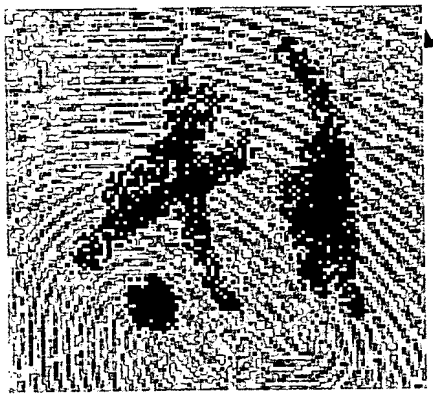


Figure 139

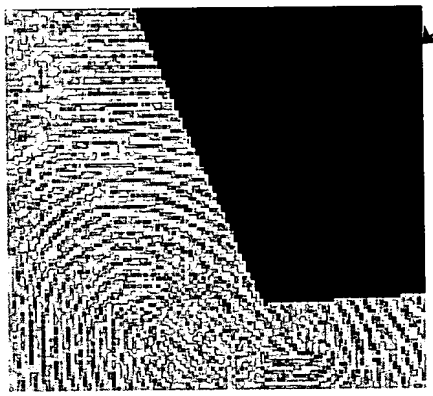


Figure 140

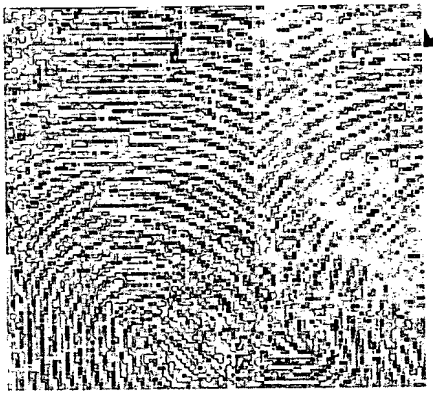


Figure 141

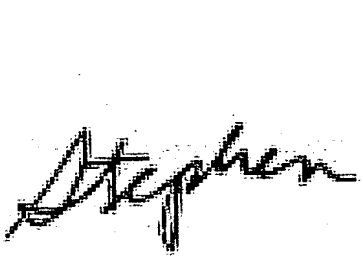


Figure 142

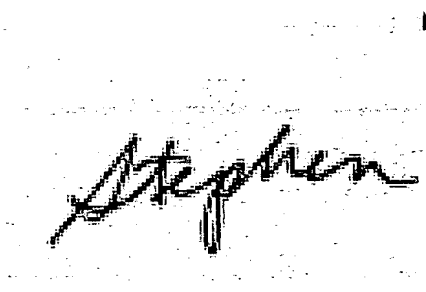


Figure 143

Figure 144

Figure 145

Figure 145

Figure 146

Figure 147

Figure 148

Figure 149

Figure 150

Figure 151

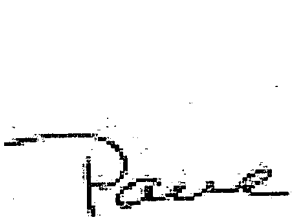


Figure 152A

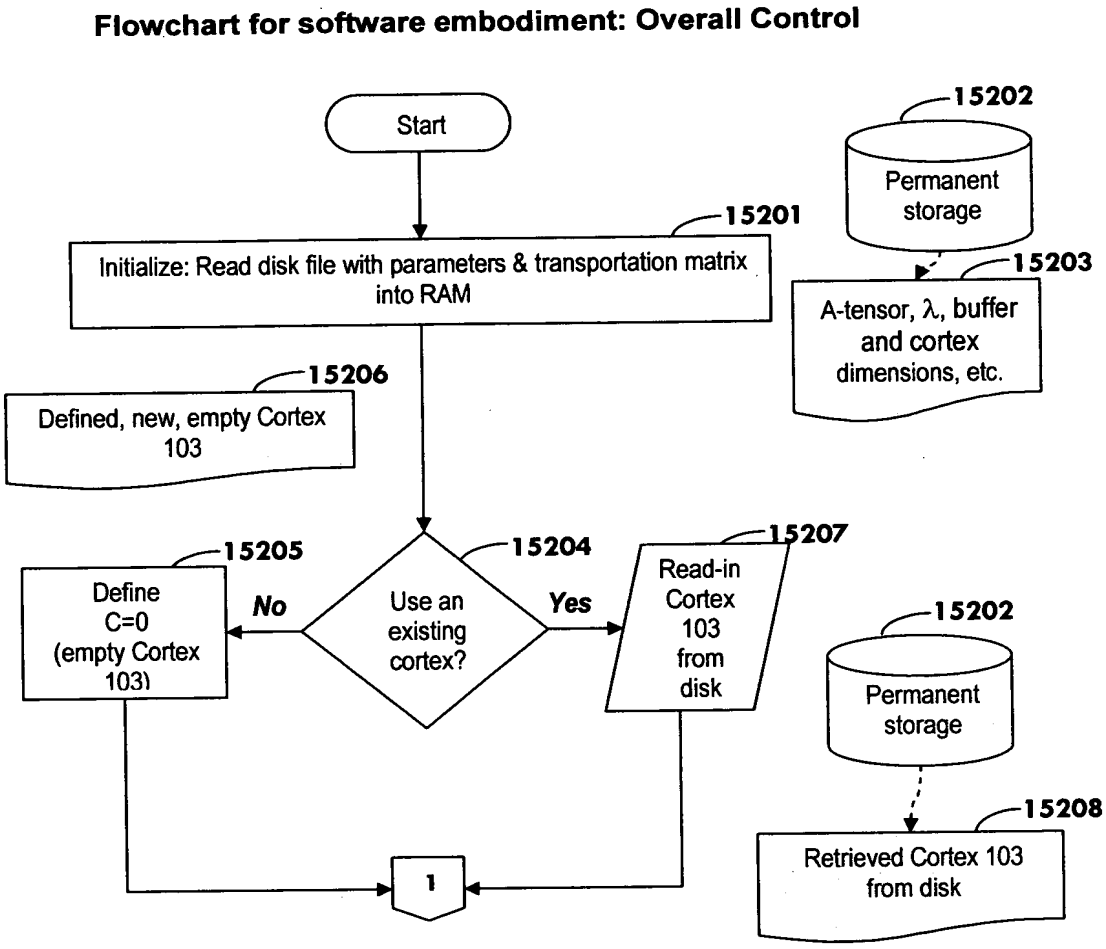

Flowchart for software embodiment: Overall Control
Start
15201
Initialize: Read disk file with parameters & transportation matrix into RAM
15202
Permanent storage
15203
A-tensor, λ, buffer and cortex dimensions, etc.
15206
Defined, new, empty Cortex 103
15205
Define C=0 (empty Cortex 103)
No
15204
Use an existing cortex?
Yes
15207
Read-in Cortex 103 from disk
15202
Permanent storage
15208
Retrieved Cortex 103 from disk
1

Figure 152B

Flowchart for software embodiment: Overall Control (continued)

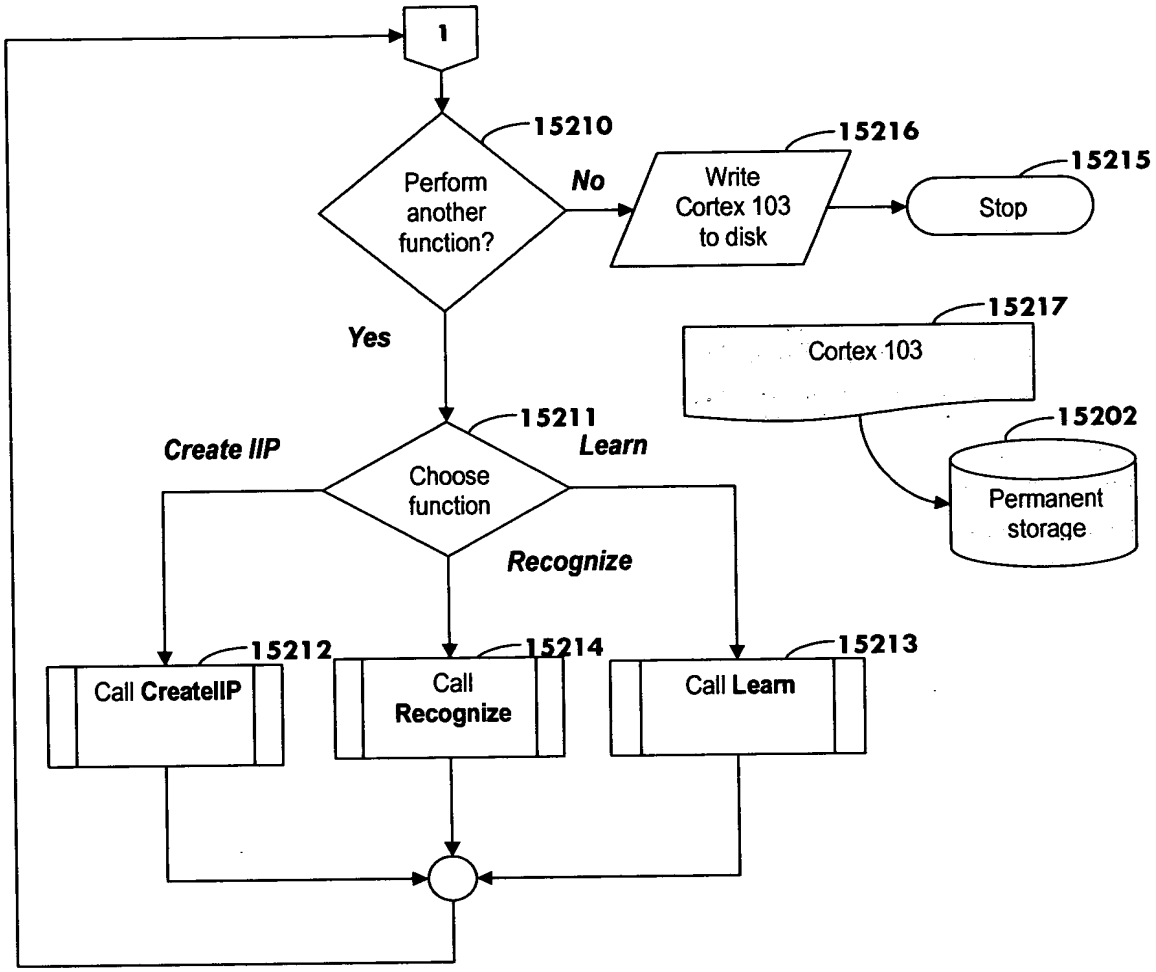


Figure 153

Flowchart for software embodiment: Create IIP

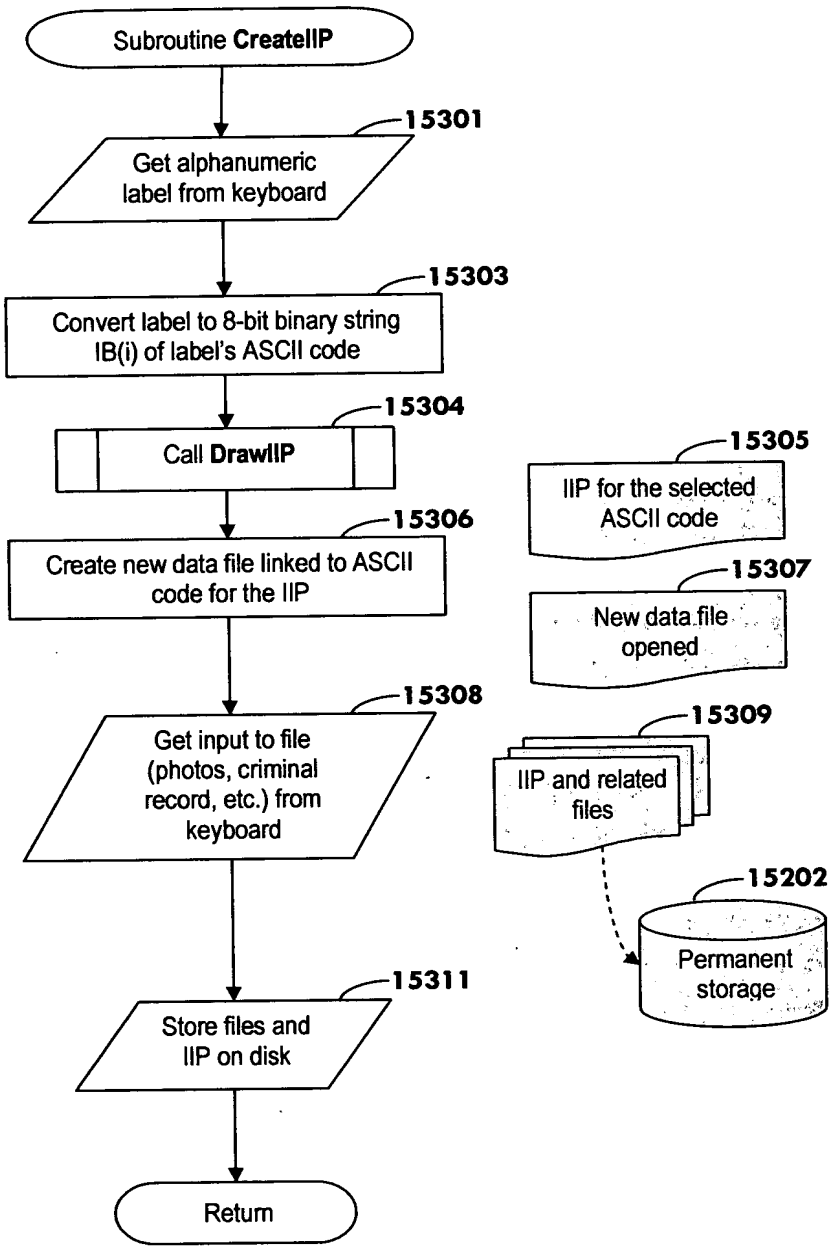


Figure 154A

Flowchart for software embodiment: Learn

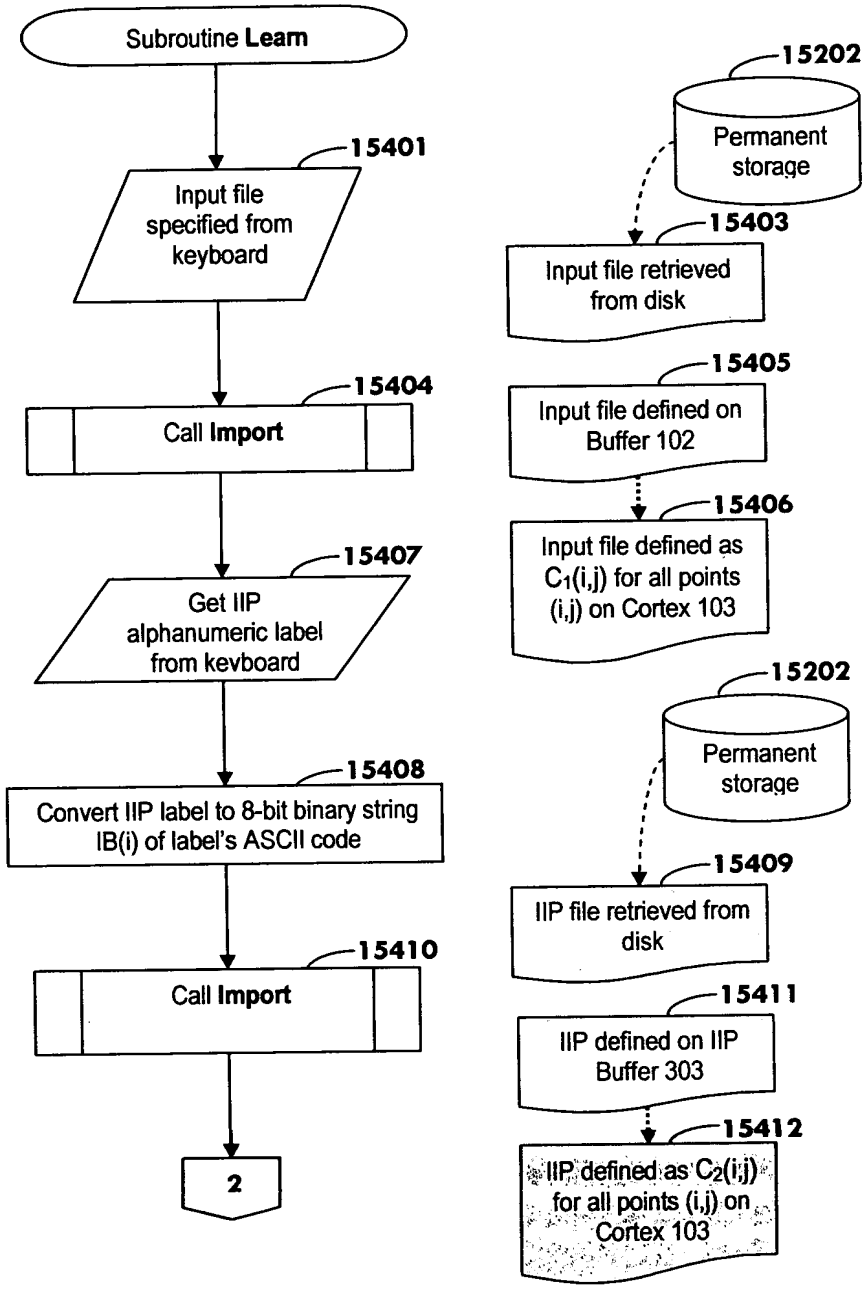


Figure 154B

Flowchart for software embodiment: Learn (continued)

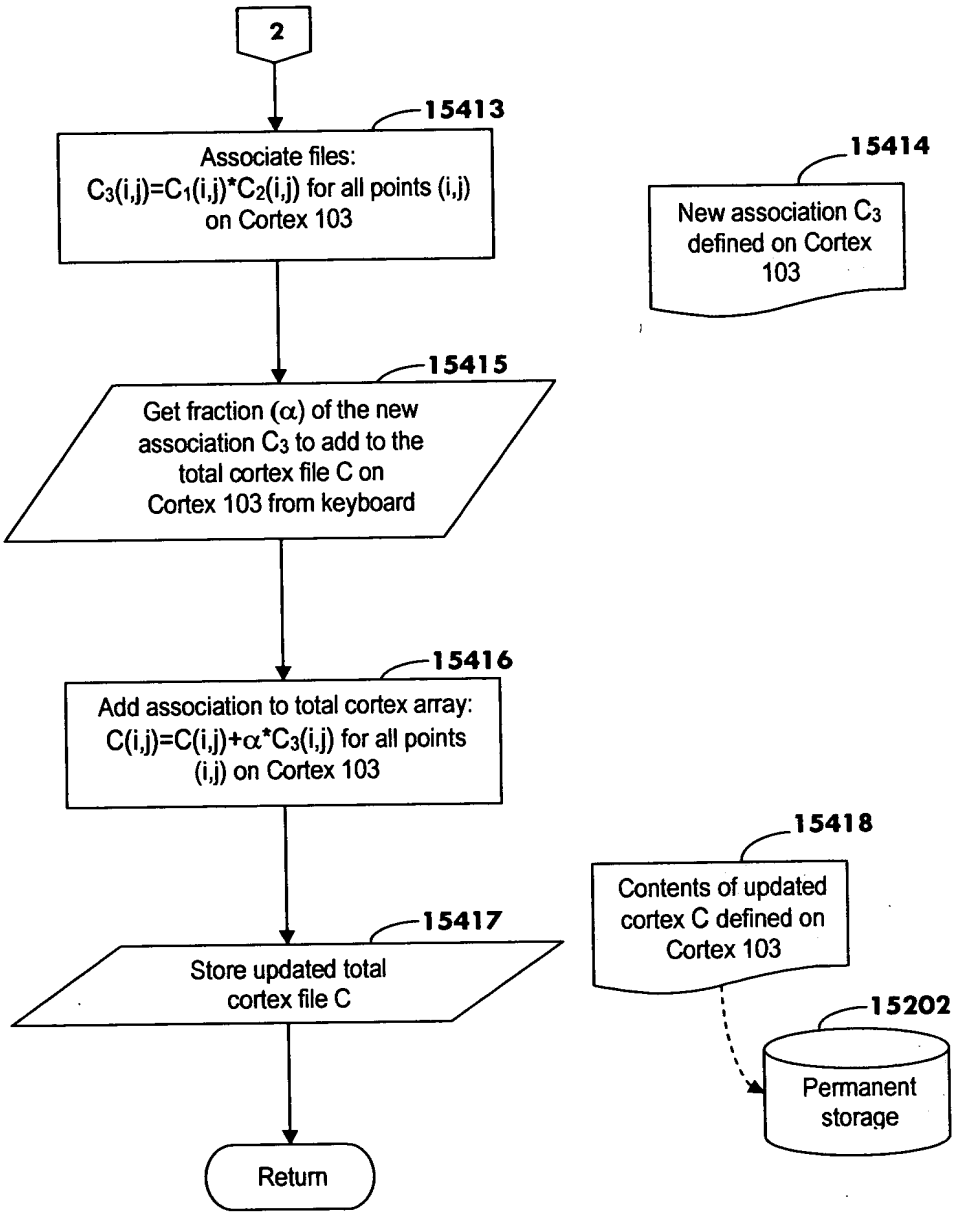


Figure 155A

Flowchart for software embodiment: Recognize

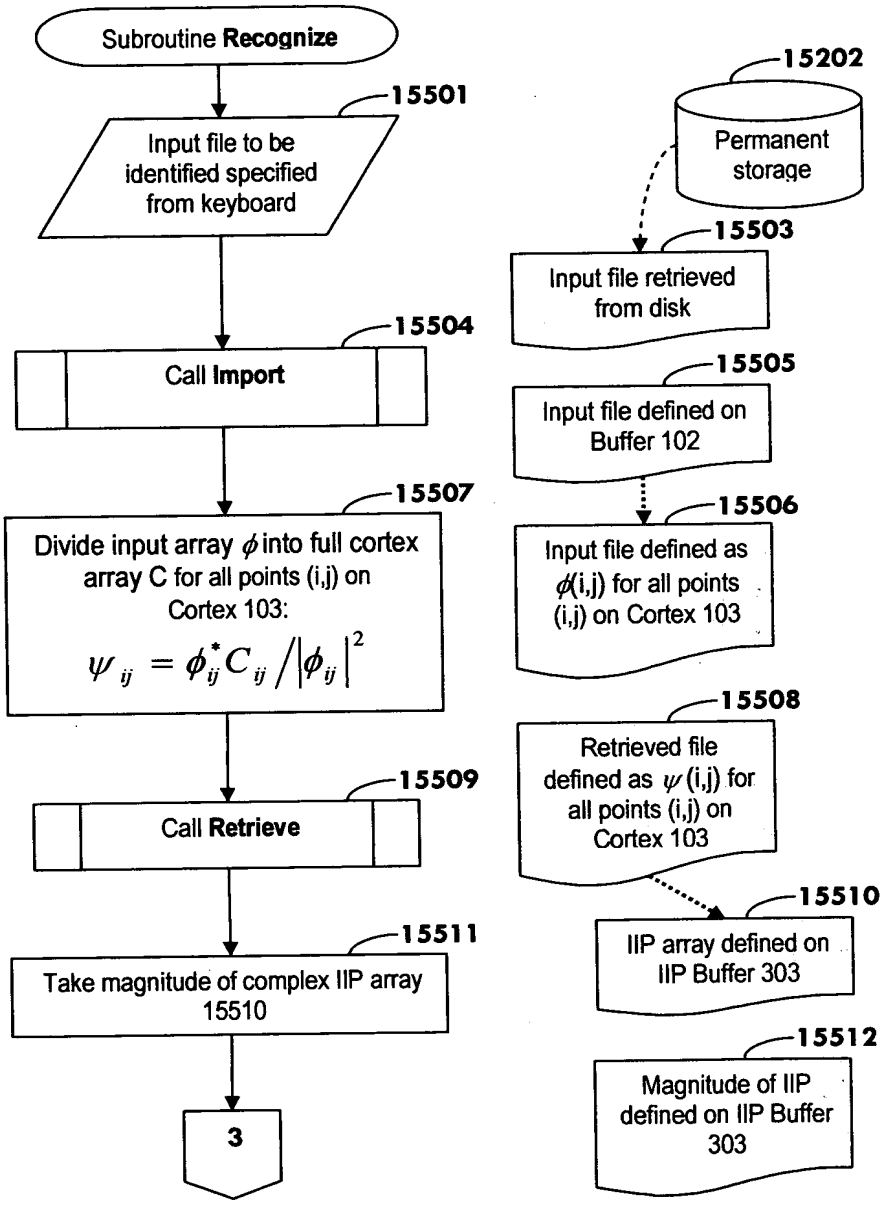


Figure 155B

Flowchart for software embodiment: Recognize (continued)

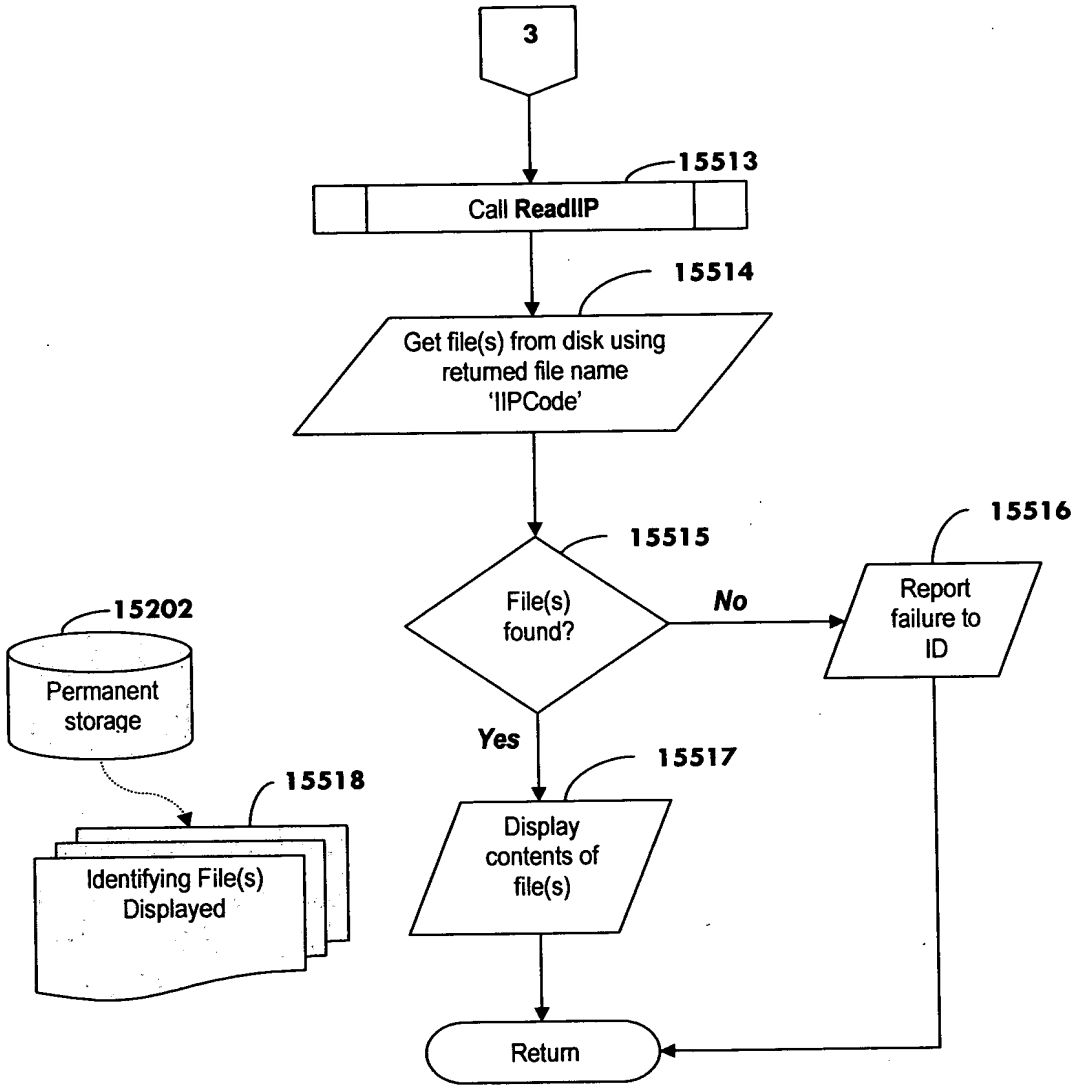


Figure 156A

Flowchart for software embodiment: Draw IIP

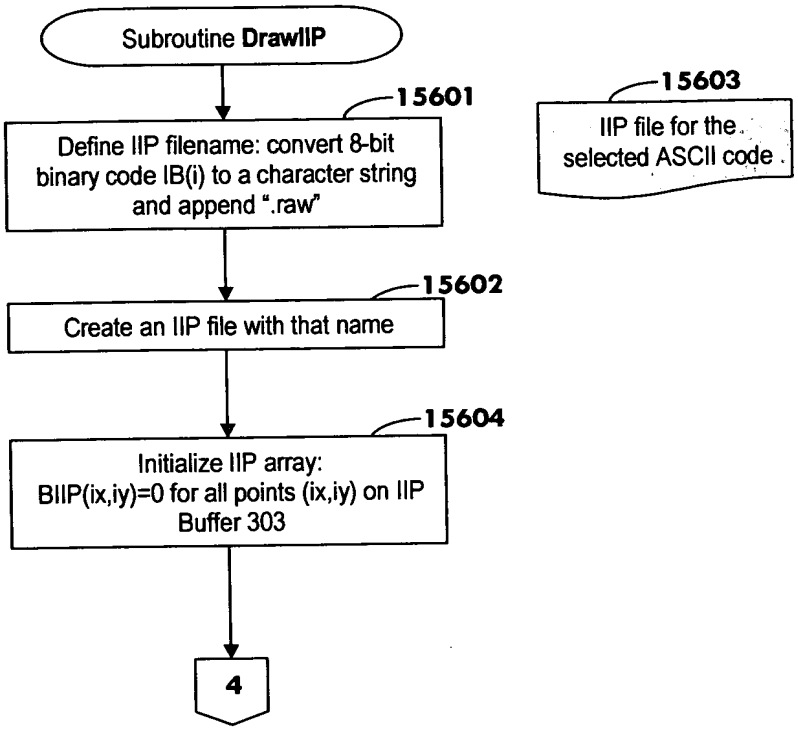


Figure 156B

Flowchart for software embodiment: Draw IIP (continued)

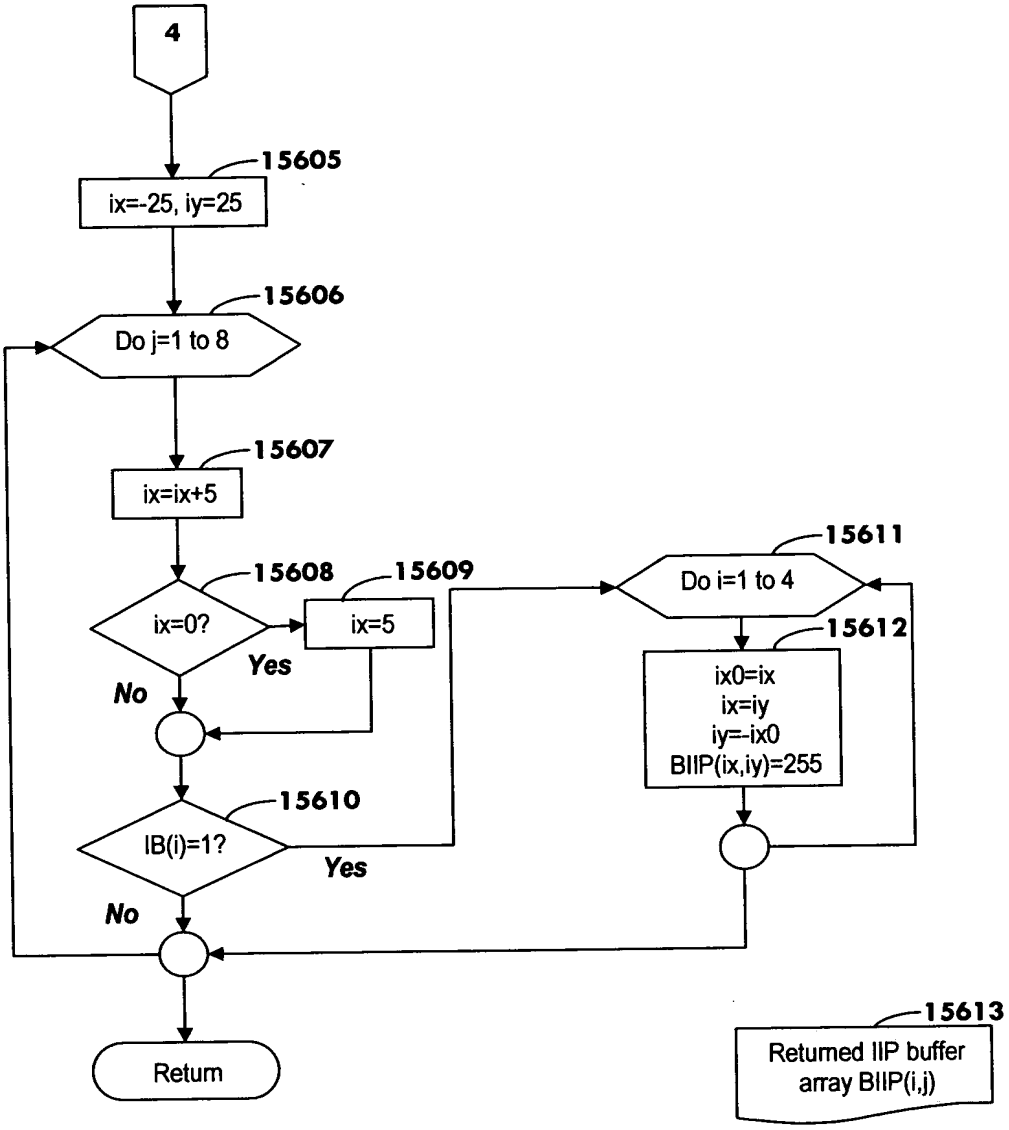


Figure 157

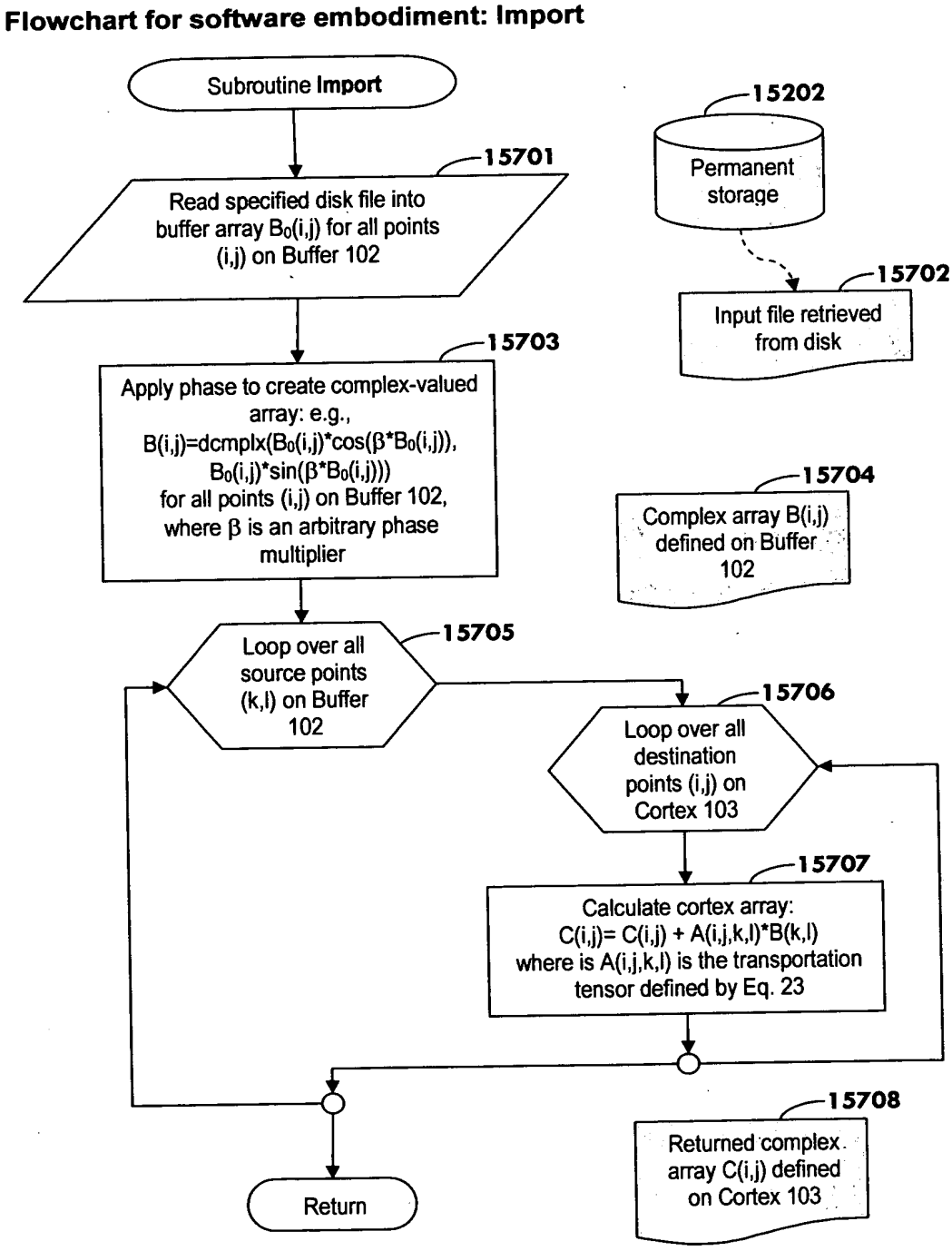

Flowchart for software embodiment: Import
Subroutine Import
15701
Read specified disk file into buffer array $B_0(i,j)$ for all points (i,j) on Buffer 102
15202
Permanent storage
15702
Input file retrieved from disk
15703
Apply phase to create complex-valued array: e.g., $B(i,j)=dcmplx(B_0(i,j)*\cos(\beta*B_0(i,j)), B_0(i,j)*\sin(\beta*B_0(i,j)))$ for all points (i,j) on Buffer 102, where β is an arbitrary phase multiplier
15704
Complex array B(i,j) defined on Buffer 102
15705
Loop over all source points (k,l) on Buffer 102
15706
Loop over all destination points (i,j) on Cortex 103
15707
Calculate cortex array: $C(i,j)= C(i,j) + A(i,j,k,l)*B(k,l)$ where is A(i,j,k,l) is the transportation tensor defined by Eq. 23
15708
Returned complex array C(i,j) defined on Cortex 103
Return

Figure 158

Flowchart for software embodiment: Retrieve

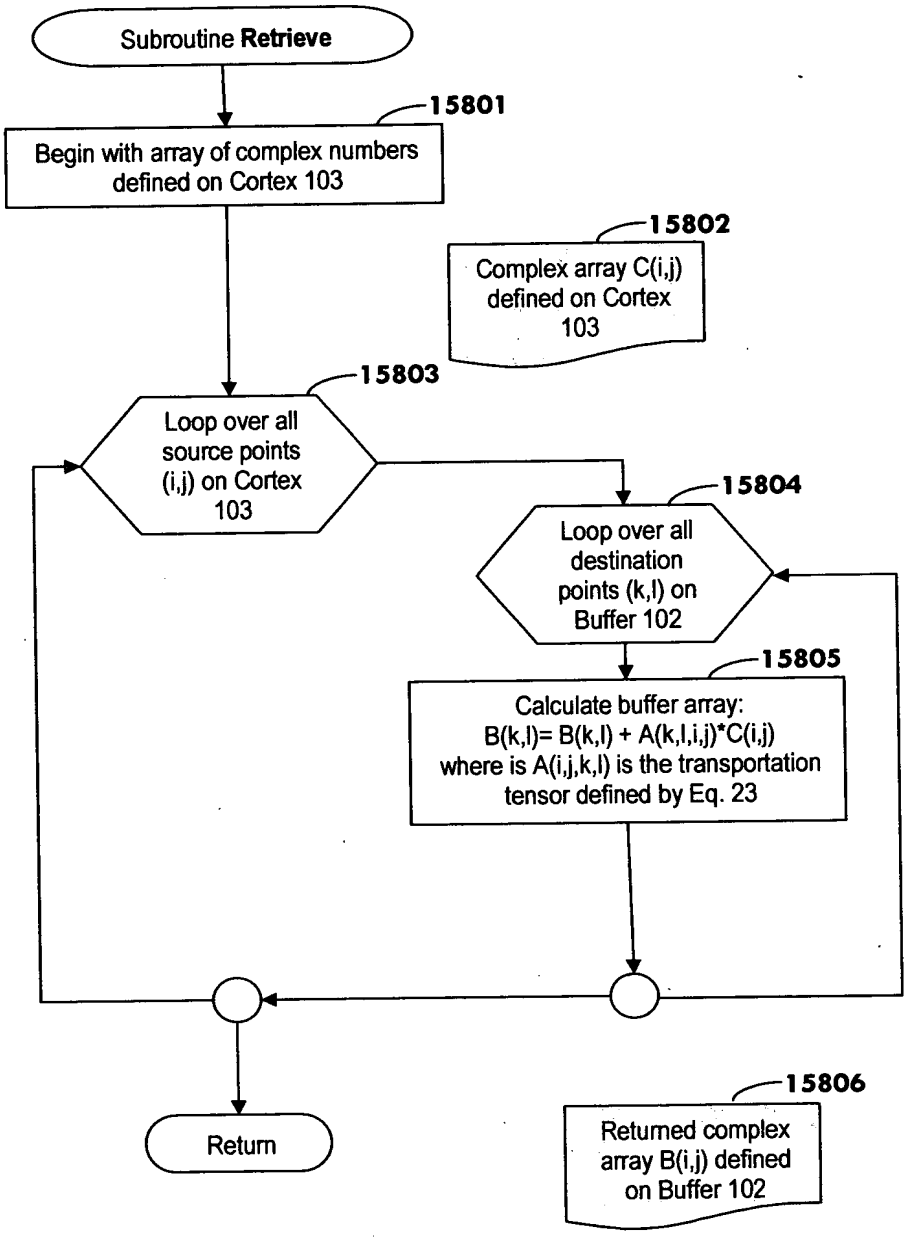


Figure 159A

Flowchart for software embodiment: Read IIP

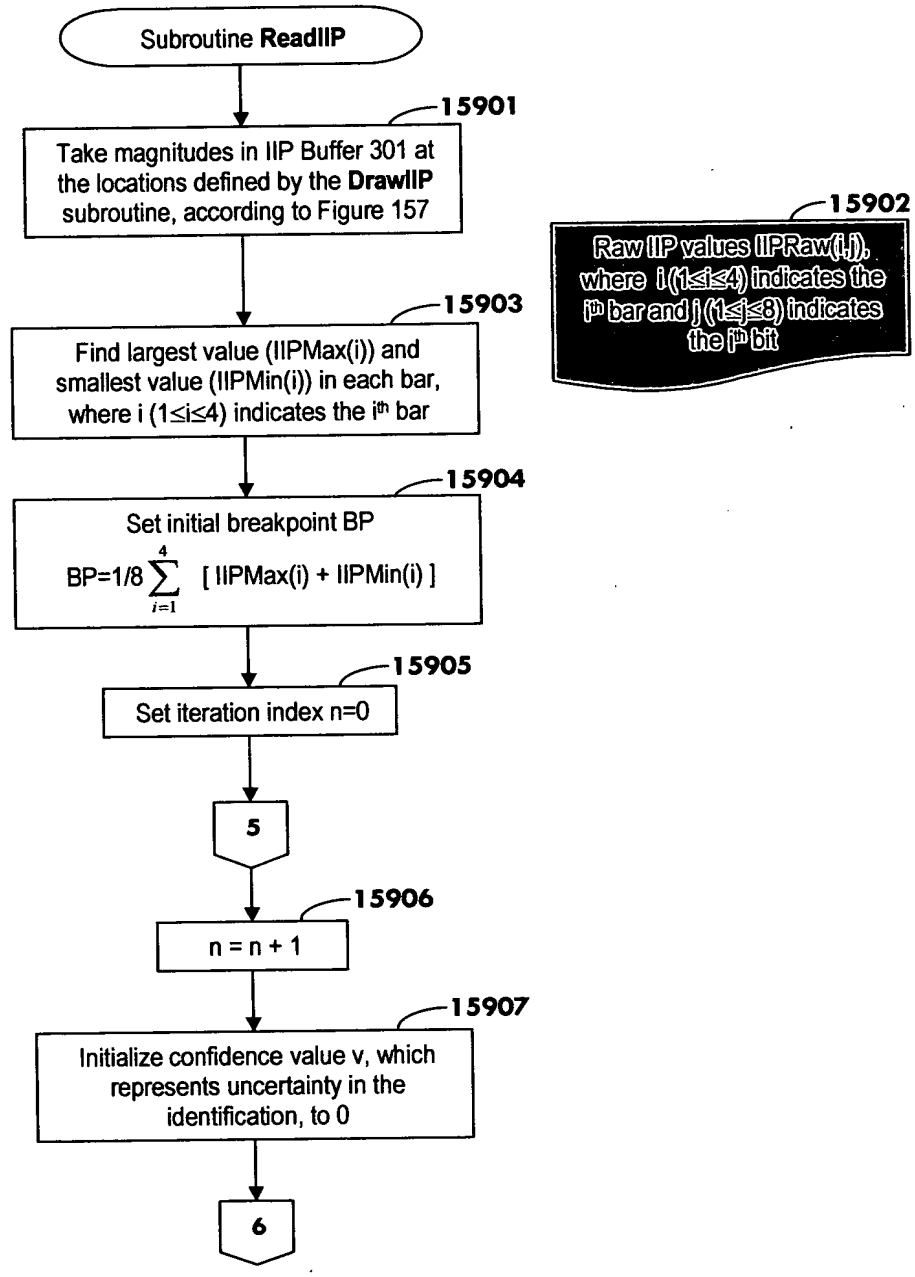


Figure 159B

Flowchart for software embodiment: Read IIP (continued)

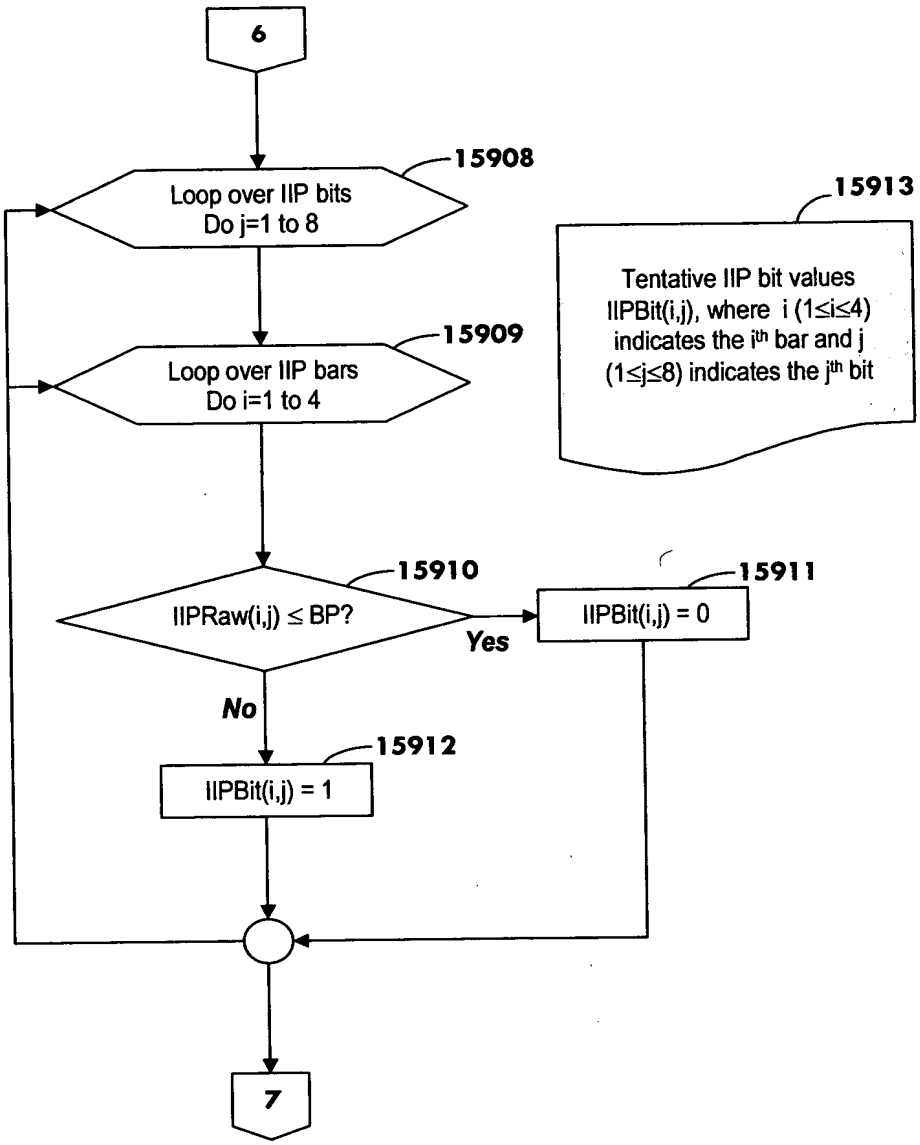


Figure 159C

Flowchart for software embodiment: Read IIP (continued)

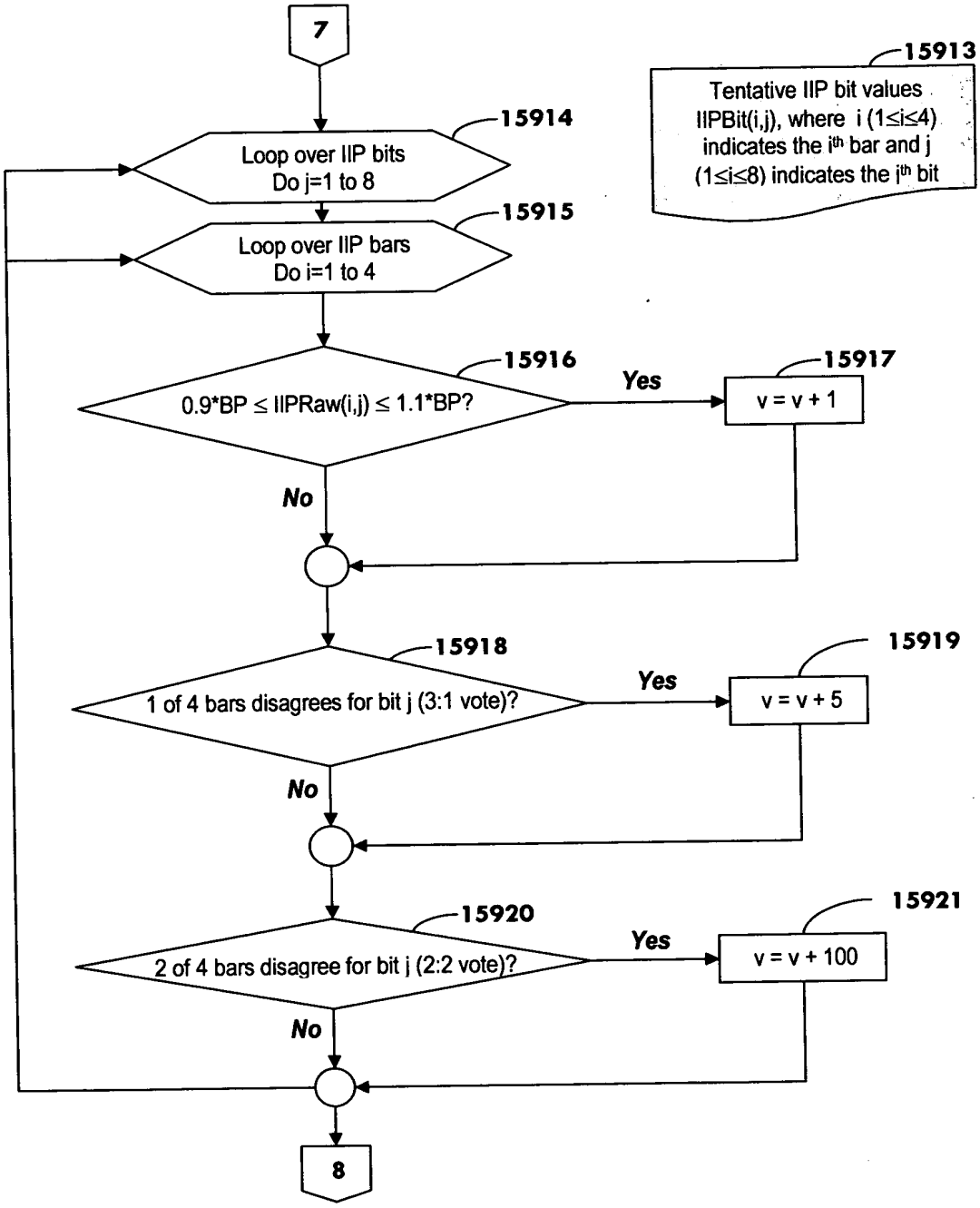


Figure 159D

Flowchart for software embodiment: Read IIP (continued)

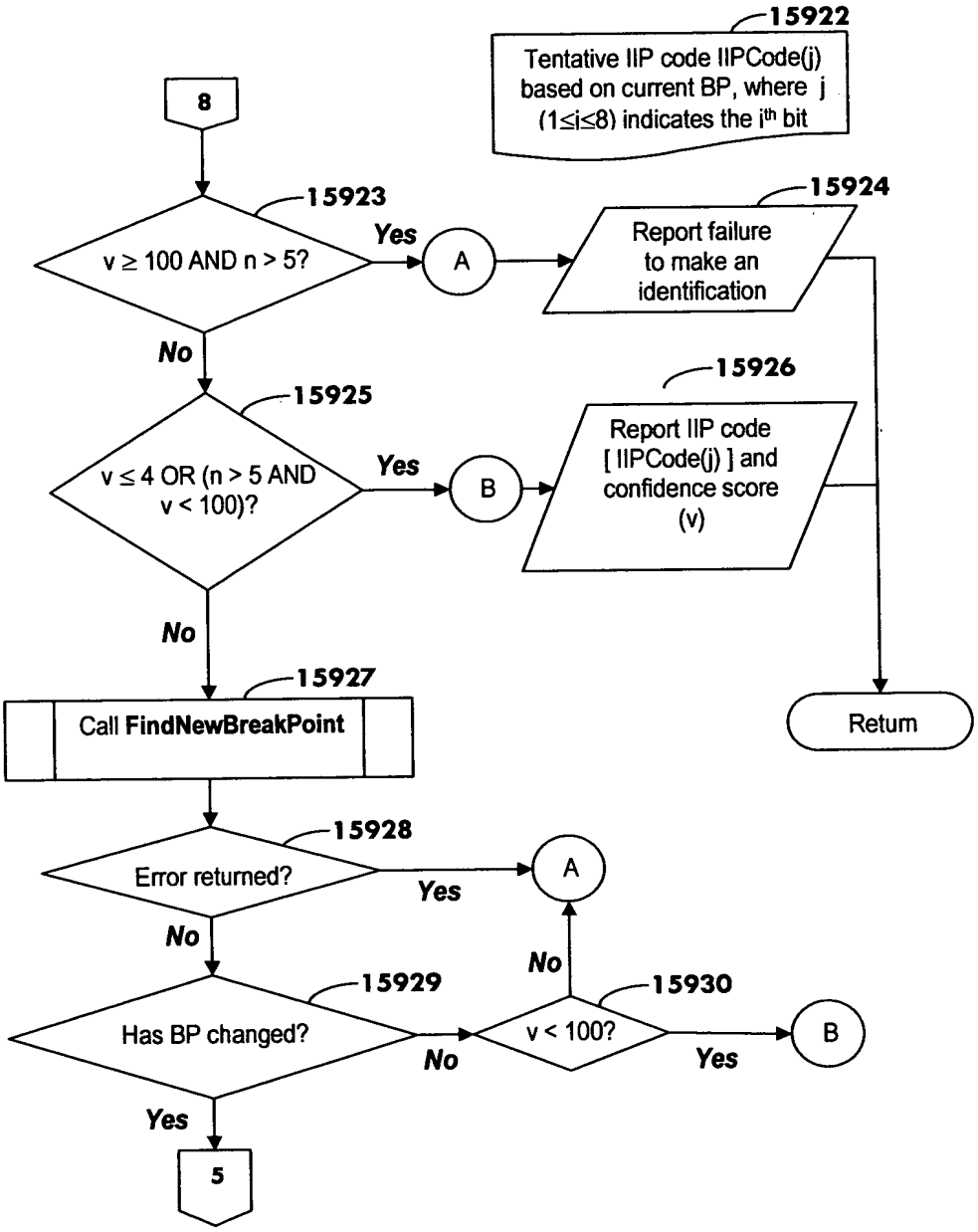
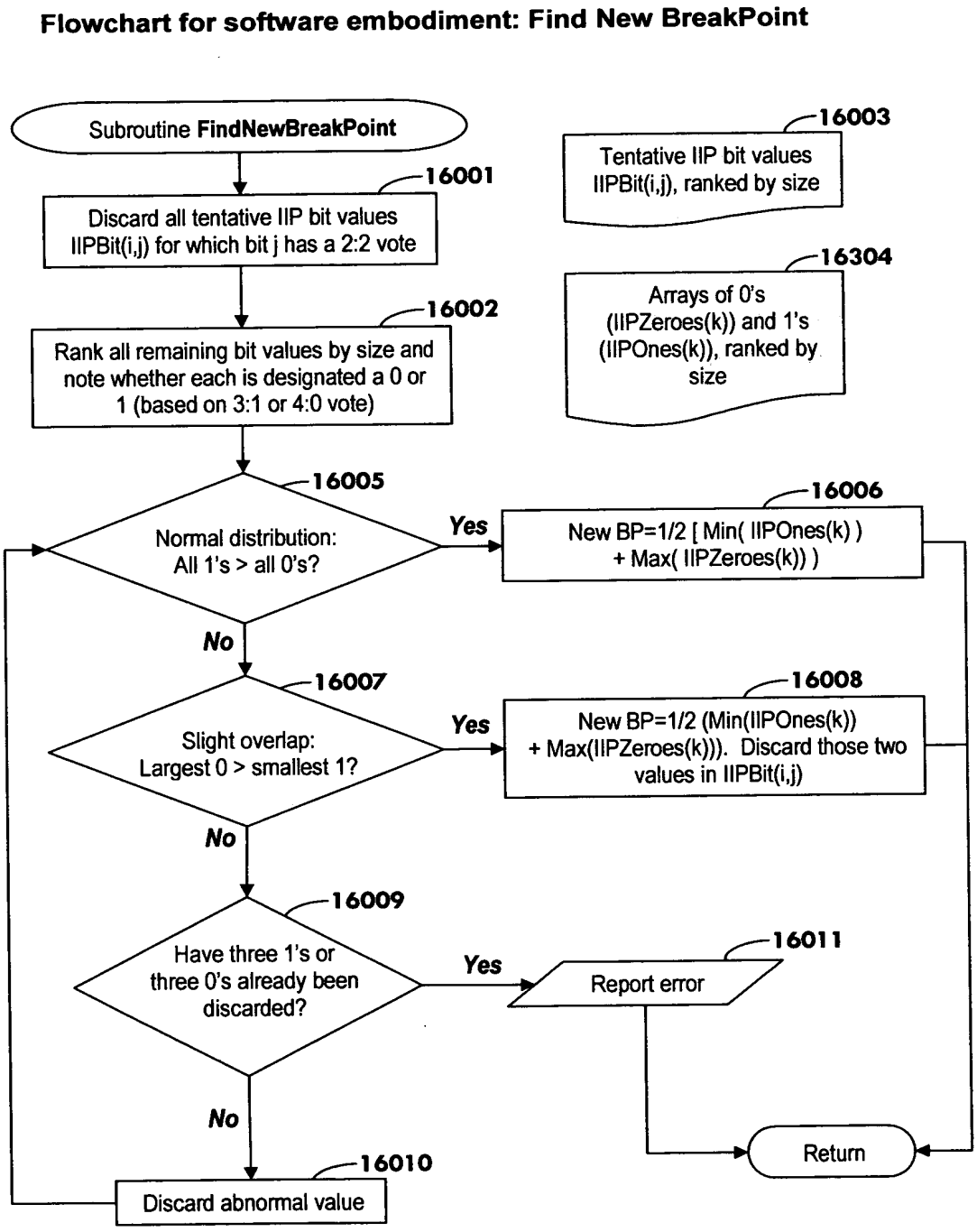


Figure 160


Flowchart for software embodiment: Find New BreakPoint
Subroutine FindNewBreakPoint
16001
Discard all tentative IIP bit values IIPBit(i,j) for which bit j has a 2:2 vote
16002
Rank all remaining bit values by size and note whether each is designated a 0 or 1 (based on 3:1 or 4:0 vote)
16003
Tentative IIP bit values IIPBit(i,j), ranked by size
16304
Arrays of 0's (IIPZeroes(k)) and 1's (IIPOnes(k)), ranked by size
16005
Normal distribution: All 1's > all 0's?
Yes
16006
New BP=1/2 [Min(IIPOnes(k)) + Max(IIPZeroes(k)))
No
16007
Slight overlap: Largest 0 > smallest 1?
Yes
16008
New BP=1/2 (Min(IIPOnes(k)) + Max(IIPZeroes(k))). Discard those two values in IIPBit(i,j)
No
16009
Have three 1's or three 0's already been discarded?
Yes
16011
Report error
No
16010
Discard abnormal value
Return